Appendices

In this supplementary material, we provide

A) Comparison with Normal Cross-Attention (Sec. A)

B) Analysis on Memory Matching Space (Sec. B)

C) Analysis on Model Complexity (Sec. C)

D) Algorithm for our LA-MB (Sec. D)

E) Visualization Analysis (Sec. E)
   - More qualitative results
   - Structural prior learned by queries
   - Failure case & Limitation

A. Comparison with Normal Cross-Attention

As discussed in the Sec. 3.1 of main paper, we deliberately design our clip-kMaX with the k-means cross-attention [7], which we empirically found to be very effective for handling the extremely large sequence of spatially and temporally flattened clip features. We now elaborate on the experiments and particularly compare with the normal (i.e., original) cross-attention [3] as well as the advanced latent memory cross-attention [1] (i.e., the cross-attention mechanism used in TubeFormer [1], which adopts latent memory to facilitate attention learning between video frames).

Tab. 1 summarizes our findings. To ensure the fairness, we employ the same backbone Axial-ResNet50-B1 [4] that has been pretrained on ImageNet-1K and Cityscapes. The baseline, employing the normal cross-attention module, yields the performance of 68.4\% STQ. The performance can be further improved by 1.6\% STQ, if we adopt the latent memory [1] in the cross-attention module. By contrast, our clip-kMaX, adopting the k-means cross-attention mechanism, attains 73.9\% STQ, significantly outperforming the conventional cross-attention and latent memory cross-attention by +5.5\% and +3.9\% STQ, respectively.

<table>
<thead>
<tr>
<th>backbone</th>
<th>method</th>
<th>STQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axial-ResNet50-B1</td>
<td>normal cross-attention (baseline)</td>
<td>68.4</td>
</tr>
<tr>
<td></td>
<td>latent memory cross-attention</td>
<td>70.0</td>
</tr>
<tr>
<td></td>
<td>k-means cross-attention (clip-kMaX)</td>
<td>73.9</td>
</tr>
<tr>
<td></td>
<td>+ LA-MB</td>
<td>74.7</td>
</tr>
</tbody>
</table>

Table 1. Comparison with Normal-Cross Attention. The k-means cross-attention adopted by our proposed clip-kMaX achieves the best STQ than the normal cross-attention and latent memory cross-attention, demonstrating the effectiveness of k-means cross-attention in video understanding task.

<table>
<thead>
<tr>
<th>size of matching space S (i.e., M x N)</th>
<th>(\tau = 1000)</th>
<th>(\tau = \text{best})</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>naive-MB</td>
<td>67.1</td>
<td>336</td>
</tr>
<tr>
<td>LA-MB</td>
<td>19.7</td>
<td>25.1</td>
</tr>
</tbody>
</table>

Table 2. Quantitative analysis on matching space size between naive-MB and our LA-MB. The size of matching space S could help us understand the difficulty of matching M objects in the memory with the detected N objects in the current frame. \(\tau\) is the hyper-parameter to refresh out the old objects. We consider two cases, where \(\tau = 1000\) to mimic the case where we rarely remove the old objects, and \(\tau = \text{best}\) uses the best hyper-parameter value for each setting.

The improvement is attributed to the effectiveness of k-means cross-attention that performs the cluster-wise argmax on cluster centers. Additionally, we show that our proposed LA-MB is complementary to clip-kMaX, which sets the best STQ performance (74.7 STQ). Our results suggest that using k-means cross-attention can reduce the ambiguity in cross-attention between queries and large flattened clip features, resulting in a higher quality of video panoptic segmentation results.

B. Analysis on Memory Matching Space

In the Sec. 3.2 of main paper, we address the limitations of the previous memory buffer approach [6], referred as
naïve-MB. One of the limitations of naïve-MB is the huge matching space in memory decoding, which increases the difficulty of matching and thus results in low association quality. From that perspective, we empirically prove that our hierarchical matching scheme, LA-MB, can effectively reduce the matching space size as shown in Tab. 2. To do so, we calculate the size of the similarity matrix $S$ (i.e., $M \times N$, where there are $M$ objects in the memory and $N$ detected objects in the current frame) to quantitatively measure the matching space size. We note that modern approaches [6] adopt a memory refreshing strategy, where the old objects stored in the memory will be removed if they are $\tau$-frame older than the current frame, which, to some degree, alleviates the issue of large matching space. However, we will show that using the memory refreshing strategy alone is not sufficient to reduce the matching space size. We compare the matching space between naïve-MB and our LA-MB under two cases of $\tau$, which is the hyper-parameter to refresh out the old objects in the memory, affecting the matching space size. In the first case, we set $\tau$ to 1000, which mimics the ideal scenario where we have a very large memory and the old objects are barely removed, aiming to exclude the effect of refreshing strategy and focus on the memory buffer approach itself. As shown in the table, we can observe that LA-MB can greatly improve the matching space efficiency by a healthy margin (i.e., 3.4× smaller and 3.6× smaller in average and max values, respectively). In the second case, $\tau$ is set to be the optimal value for each memory buffer approach (i.e., 1 for naïve-MB and 10 for LA-MB). As shown in the table, the memory refreshing strategy effectively reduces the matching space size for naïve-MB. However, our LA-MB still outperforms naïve-MB by achieving 9.1× and 8.2× more efficient matching space in average and max value, respectively.

![Query Visualization on KITTI-STEP val set](image)

**Figure 1. Query Visualization** on KITTI-STEP val set. We use Video-kMaX with Axial-ResNet50-B1 backbone that is trained on KITTI-STEP and then plot the location of averaged mask center (including all stuff and things) predicted by each query.

### C. Analysis on Model Complexity

In Tab. 3, we measure params, FLOPs, and FPS for our method, using a Tesla V100 GPU with CUDA 11.0 and batch size 1. We run the inference 3 times to obtain the average and worst FPS.

### D. Algorithm for our LA-MB

In Alg. 1, we provide the algorithm for our Location-Aware Memory Buffer (LA-MB), which 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. For better understanding, we also attach our code snippet in the supplementary materials.
Algorithm 1: Algorithm for LA-MB

Input:
1. LA-MB = \{(\hat{q}_i^{t-1}, \hat{b}_i^{t-1})\}_{j=1}^{M} with M encoded objects until frame t − 1.
2. Feature set (q_i, b_i) of object i in current frame t.
3. Panoptic map \( P \in \mathbb{R}^{H \times W} \) of previous frame \( t - 1 \).

Output: (updated ID of object i, updated LA-MB)

\begin{align*}
\textbf{begin} & \quad \# Decoding phase. \\
\textbf{if} & \quad \text{Video-Stitch(object i and P) \leq M} \quad \textbf{then} \\
& \quad k \leftarrow \text{Video-Stitch(object i and P)} \\
\textbf{else} & \\
& \quad f(i, j) = e^{-\|b_i - b_j\|^2/2 \cdot \cos(q_i, \hat{q}_i)} \\
& \quad r = \arg \max_{M} \left\{ f(i, j)_{j=1}^{M} \right\} \\
& \quad \textbf{if} \ f(i, r) \geq \alpha \quad \textbf{then} \\
& \quad \quad k \leftarrow r \\
& \quad \quad \textbf{else} \\
& \quad \quad \quad \quad \quad k \leftarrow M + 1 \\
\textbf{end} & \quad \# Encoding phase. \\
\textbf{if} & \quad k \leq M \quad \textbf{then} \\
& \quad \# The object is tracked in the memory. \\
& \quad \hat{q}_k^t = (1 - \lambda) \hat{q}_k^{t-1} + \lambda q_i \\
& \quad \hat{b}_k^t = b_i \\
& \quad \textbf{for} \ j \in \{1, \ldots, k - 1, k + 1, \ldots, M\} \textbf{ do} \\
& \quad \quad \hat{q}_j^t = \hat{q}_j^{t-1} \\
& \quad \quad \hat{b}_j^t = \hat{b}_j^{t-1} + (\hat{b}_j^{t-1} - \hat{b}_j^{t-2}) \\
\textbf{else} & \\
& \quad \# The object is new. \\
& \quad \hat{q}_i^t = q_i \\
& \quad \hat{b}_i^t = b_i \\
& \quad \textbf{for} \ j \in \{1, \ldots, M\} \textbf{ do} \\
& \quad \quad \hat{q}_j^t = \hat{q}_j^{t-1} \\
& \quad \quad \hat{b}_j^t = \hat{b}_j^{t-1} + (\hat{b}_j^{t-1} - \hat{b}_j^{t-2}) \\
\textbf{return} & \quad (k, \text{updated LA-MB})
\end{align*}

E. Visualization Analysis

More qualitative results We show some visualization results in Fig. 2 for VIPSeg, where the baseline naïve-MB fails to associate persons in a crowd, since they have similar appearance features. On the other hand, our LA-MB correctly associates the same person by effectively exploiting both the appearance and location features. In our supplementary submission, we also include video panoptic segmentation results on the validation sets of KITTI-STEP [5] and VIPSeg [2]. Our Video-kMaX (consisting of clip-kMaX and LA-MB) demonstrates more clear and consistent video results than the baselines.

Structural prior learned by queries We observe that the object queries learned by our Video-kMaX demonstrate a structural prior that a particular query will respond to objects around a specific location on the image plane. To visualize the structural prior, for each query, we compute the mean location center of all its segmented objects in the whole KITTI-STEP validation set, and show the scatter plot in Fig. 1. As shown in the figure, each object query is responsible to segment objects around a specific location on the image plane. Interestingly, the object queries are scattered mostly along a vertical and a horizontal line, showing the property of ego-centric car in KITTI-STEP, where the street-view images are collected by a driving car.

Failure case and Limitation We analyze the failure mode of our Video-kMaX in Fig. 3. The first row and second row are video frames and corresponding video panoptic results with our Video-kMaX, respectively. We observe that a person initially assigned with ID number 99 until frame 2 is re-assigned with different ID numbers, i.e., 107 (in frame 3) and 108 (in frame 4). The ID switch could be attributed to two reasons. First, the appearance feature of the occluded person (i.e., person ID 107 in frame 3) is not reliable, as most of its discriminative appearance regions are occluded. Second, the target object demonstrates a large random movement, violating our slow linear motion assumption encoded by the location feature. This failure case presents a challenging but interesting research direction to further improve our model by strengthening both appearance and location features.

References


Figure 2. Visualization results on VIPSeg \textit{val} set. The baseline naive-MB, only exploiting the appearance feature, fails to associate the same person, as neighboring people have similar appearance features. On the other hand, our LA-MB, exploiting both appearance and location features, successfully associates the same person.

Figure 3. Failure case on VIPSeg \textit{val} set. The target object is initially assigned with ID 99. Its ID switches to 107 and 108 in frame 3 and frame 4, respectively. Our method fails to track the target object, because it is heavily occluded and moves at a large random pace, making both appearance and location features unreliable.

