Hierarchical Visual Prompt Learning for Continual Video Instance Segmentation

Supplementary Material

A. Experimental Settings

Benchmark Datasets: We conduct comparison experiments on three video instance segmentation datasets (*i.e.*, YouTube-VIS 2019[*] [77], YouTube-VIS 2021[†] [77] and OVIS[‡] [55]) to verify the effectiveness of our HVPL model in addressing the CVIS problem. **YouTube-VIS 2019** [77] is the first dataset proposed to perform video instance segmentation and it has 40 semantic categories. we consider 20-2 and 20-5 settings on YouTube-VIS 2019. The 20-2 and 20-5 settings respectively indicate first learning 20 classes, followed by ten continual tasks, each with 2 new classes ($T = 11$); and followed by 4 consecutive tasks, each with 5 new classes ($T = 5$). **YouTube-VIS 2021** [77] comprises the same semantic categories with YouTube-VIS 2019, but has more confusing trajectories than YouTube-VIS 2019. On YouTube-VIS 2021, we set 30-10 and 20-4 under the CVIS setting. The settings of 30-10 and 20-4 involve learning 30 classes followed by one task with 10 new classes ($T = 2$), and learning 20 classes followed by 5 tasks with 4 new classes each ($T = 6$). **OVIS** [55], which contains 25 classes, has distinct characteristics compared to YouTube-VIS 2021 [77]: each video features more instances with heavy occlusions and diverse appearances. On OVIS [55], we set 15-5 and 15-10 for the CVIS problem. The settings of 15-5 and 15-10 involve learning 15 classes followed by 2 task with 5 new classes each ($T = 3$), and learning 15 classes followed by one task with 10 new categories ($T = 2$).

Implementation Details: For the network architecture, we introduce Mask2Former [10] as the frame-level detector, where the backbone is ResNet-50 [32], the Transformer decoder $\mathcal{D}_{\text{trans}}$ includes 9 MSA layers, and the number of scales in the pixel decoder $\mathcal{D}_{\text{pixel}}$ is 3. Besides, the video context decoder $\mathcal{D}_{\text{video}}$ consists of 6 GSS layers and 3 MSA layers. All network parameters of the pretrained Mask2Former [10] are frozen during the training phase. Following [16], we set the feature dimensions as $D = 256$ and the number of heads in each MSA layer as 8. For the lengths of task-specific frame and video prompts, we set $L_p^f = 100, L_p^v = 100$ for the first VIS task and set $L_p^f = 10, L_p^v = 10$ to learn the t-th ($t \geq 2$) incremental task. Furthermore, we adopt the same preprocessing strategy for input video frames during training and inference as proposed in [33]. We optimize the parameters of the video context decoder $\mathcal{D}_{\text{video}}$ at the first task, and freeze them when $t \geq 2$. If the learning of the $(t{-}1)$-th task is completed, we will store the feature space $\mathcal{O}^{t-1}$ in advance, and delete the feature space $\mathcal{O}^{t-2}$ learned at the $(t{-}2)$-th task for saving memory costs. Then $\mathcal{O}^{t-1}$ is utilized to learn the t-th task. To balance the forgetting of old tasks and the learning of new tasks, we empirically set $\xi = 0.7$ to determine the orthogonal feature space $\widehat{\mathbf{V}}_0^{t-1}$ of the $(t{-}1)$-th task. Additionally, we utilize the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) to optimize the proposed HVPL model, where the initial learning rate is 5.0×10^{-5}. The random seed is set to 42 in this paper.

```
Algorithm 2: Optimization of The Proposed HVPL.
Initialize: The pretrained Mask2Former and a sequence of
  video instance segmentation tasks $\mathcal{T} = \{\mathcal{T}^t\}_{t=1}^T$;
▷ Training for The $t$-th Task:
Initialize the frame and video prompts $\{\mathbf{P}_{\text{frm}}^t, \mathbf{P}_{\text{vid}}^t\}$;
Initialize the classifier and mask heads $\{\Gamma_c^t, \Gamma_m^t\}$;
for $(\mathbf{x}_i^t, \mathbf{y}_i^t)$ in $\mathcal{T}^t$ do
    if $t \geq 2$ then
        Update task-specific video prompt $\mathbf{P}_{\text{vid}}^t$, classifier
          and mask heads $(\Gamma_c^t, \Gamma_m^t)$ via the loss in [33];
        Compute the original gradient $\triangle\mathbf{P}$ used to update
          the task-specific frame prompt $\mathbf{P}_{\text{frm}}^t$;
        Conduct SVD on the feature space $\mathcal{O}^{t-1}$;
        Obtain the orthogonal feature space $\widehat{\mathbf{V}}_0^{t-1}$;
        Obtain $\triangle\mathbf{P}^*$ via gradient projection in Eq. (11);
        Use $\triangle\mathbf{P}^*$ to update the frame prompt $\mathbf{P}_{\text{frm}}^t$;
    else
        Update learnable parameters via the loss in [33];
Delete the feature space $\mathcal{O}^{t-1}$ of the $(t{-}1)$-th task;
Construct the feature space $\mathcal{O}^t$ for the $t$-th task;
Return: $\{\mathbf{P}_{\text{frm}}^t, \mathbf{P}_{\text{vid}}^t, \Gamma_c^t, \Gamma_m^t, \mathcal{O}^t\}$.
▷ Inference:
Initialize: All task-specific frame and video prompts
  $\{\mathbf{P}_{\text{frm}}^t, \mathbf{P}_{\text{vid}}^t\}_{t=1}^T$, along with the classifier and mask
  heads $\{\Gamma_c^t, \Gamma_m^t\}_{t=1}^T$, learned so far;
Obtain $\mathbf{P}_{\text{frm}}^* = [\mathbf{P}_{\text{frm}}^1; \mathbf{P}_{\text{frm}}^2; \cdots; \mathbf{P}_{\text{frm}}^T] \in \mathbb{R}^{TL_p^f \times D}$;
Obtain $\mathbf{P}_{\text{vid}}^* = [\mathbf{P}_{\text{vid}}^1; \mathbf{P}_{\text{vid}}^2; \cdots; \mathbf{P}_{\text{vid}}^T] \in \mathbb{R}^{TL_p^v \times D}$;
Obtain $\Gamma_c^* = [\Gamma_c^1, \Gamma_c^2, \cdots, \Gamma_c^T] \in \mathbb{R}^{D \times (|\mathcal{Y}^{1:t}|+1)}$;
Return: Predictions about masks and class probabilities.
```

Comparative Methods: To exhibit the effectiveness of the proposed HVPL model, we introduce some state-of-the-art continual learning methods for comparison experiments. Specifically, MiB [6] devises a novel objective function alongside a tailored classifier initialization strategy to address the issue of background shift. CoMFormer [7] utilizes a new adaptive distillation loss combined with a mask-based pseudo-labeling technique to effectively mitigate forgetting. NeST [76] proposes a classifier pretuning method, which is applied prior to the formal training process. Instead of

[*] https://codalab.lisn.upsaclay.fr/competitions/6064

[†] https://codalab.lisn.upsaclay.fr/competitions/7680

[‡] https://codalab.lisn.upsaclay.fr/competitions/4763

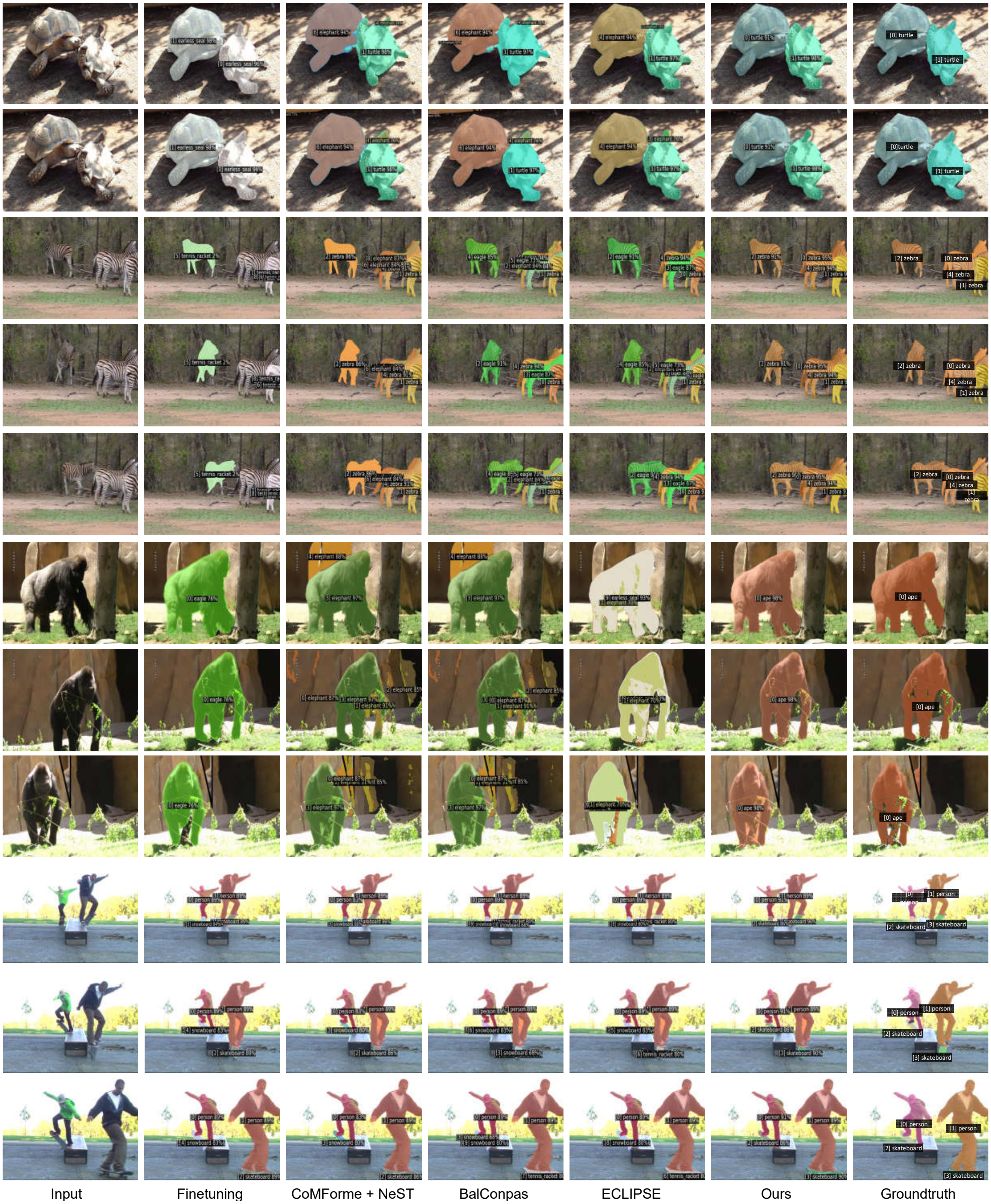

Figure 5. Comparison results of some selected video frames from YouTube-VIS 2019 [77] under the 20-5 setting (zoom in for a better view).

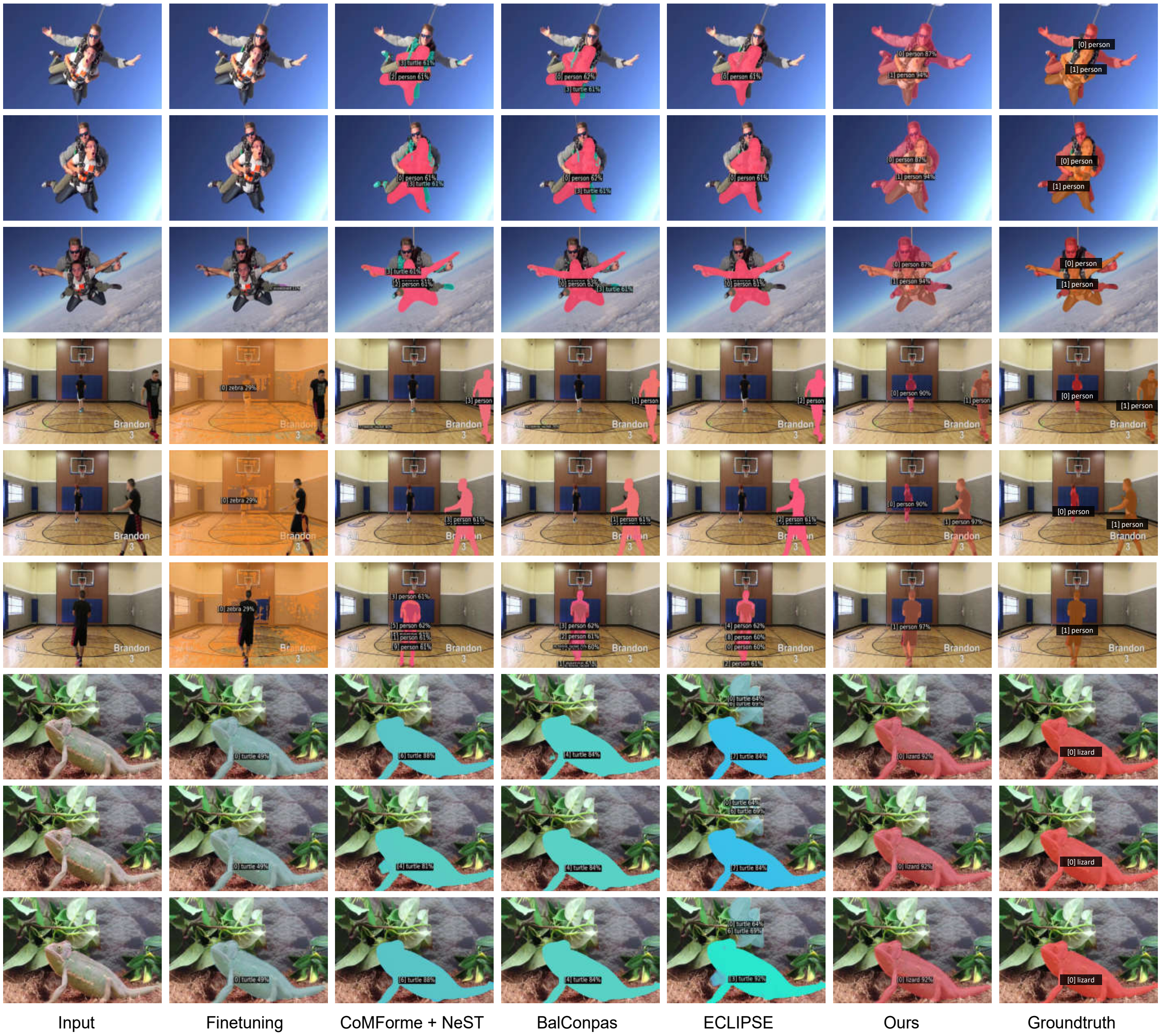

Figure 6. Comparisons of some selected video frames from YouTube-VIS 2021 [77] under the 30-10 setting (zoom in for a better view).

directly adjusting the parameters of new classifiers, NeST learns a transformation from old classifiers to generate new classifiers for initialization. BalConpas [8] introduces a class-proportional memory strategy that ensures the class distribution in the replayed sample set aligns with the distribution in the historical training data. ECLIPSE [41] devises an efficient method for continual panoptic segmentation built on visual prompt tuning. For fair comparisons, all comparative methods employ the same backbone and data augmentation strategy for training.

B. Optimization

The optimization pipeline of our HVPL model is presented in **Algorithm** 2. During training, given a vide-label pair $(\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathcal{T}^t$ at the t-th ($t \geq 2$) task, we first update the task-specific video prompt $\mathbf{P}^t_{\text{vid}}$, the classifier and mask heads (Γ_c^t, Γ_m^t) via optimizing the loss proposed in [33]. Then we compute the original gradient $\triangle\mathbf{P}$ used to update the task-specific frame prompt $\mathbf{P}^t_{\text{frm}}$, and conduct SVD on the feature space $\mathcal{O}^{t-1}$ of the $(t-1)$-th task. After obtaining the orthogonal feature space $\widehat{\mathbf{V}}_0^{t-1}$, we perform gradient projection to derive $\triangle\mathbf{P}^* = \triangle\mathbf{P}\widehat{\mathbf{V}}_0^{t-1}(\widehat{\mathbf{V}}_0^{t-1})^\top$ via Eq. (11), and employ $\triangle\mathbf{P}^*$ to update the task-specific frame prompt $\mathbf{P}^t_{\text{frm}}$. During inference, we first concatenate all frame prompts $\{\mathbf{P}^t_{\text{frm}}\}_{t=1}^T$ as $\mathbf{P}^*_{\text{frm}} = [\mathbf{P}^1_{\text{frm}}; \mathbf{P}^2_{\text{frm}}; \cdots; \mathbf{P}^T_{\text{frm}}] \in \mathbb{R}^{TL_p^f \times D}$, all task-specific video prompts $\{\mathbf{P}^t_{\text{vid}}\}_{t=1}^T$ as $\mathbf{P}^*_{\text{vid}} = [\mathbf{P}^1_{\text{vid}}; \mathbf{P}^2_{\text{vid}}; \cdots; \mathbf{P}^T_{\text{vid}}] \in \mathbb{R}^{TL_p^v \times D}$, and all classifier heads $\{\Gamma_c^t\}_{t=1}^T$ as $\Gamma_c^* = [\Gamma_c^1, \Gamma_c^2, \cdots, \Gamma_c^T] \in \mathbb{R}^{D\times(|\mathcal{Y}^{1:t}|+1)}$. Subsequently, we utilize them to predict masks and class probabilities for a given test video.

C. Qualitative Comparisons

As shown in Figs. 5–6, to evaluate the performance of our proposed model in various CVIS scenarios, we present the visualization results of selected video frames from YouTube-VIS 2019 and YouTube-VIS 2021 [77]. The following observations can be drawn from the results: 1) Significant Improvement Over Prompt Learning-Based Methods: The proposed model outperforms the prompt learning-based method ECLIPSE [41] across all settings, demonstrating superior capability in addressing CVIS problem. Notably, in complex backgrounds and dynamic scenarios, our model mitigates catastrophic forgetting at both the frame and video levels. 2) Better Performance Than Knowledge Distillation-Based Methods: Compared to knowledge distillation methods such as CoMFormer [7], NeST [76] and BalConpas [8], our model achieves more precise instance segmentation, especially in cases involving multiple or small objects. This highlights the effectiveness of the task-specific video prompt and the video context decoder in capturing global video contexts to tackle catastrophic forgetting. These visual results further validate the effectiveness of the proposed model in tackling forgetting of old classes at both the frame and video levels.

D. Societal Impact and Limitations

Societal Impact: Continual Video Instance Segmentation (CVIS) is an emerging research field at the intersection of computer vision and continual learning, focusing on the ability to segment, track, and incrementally learn from objects in video streams over time. Unlike traditional video instance segmentation, which operates on fixed, predefined datasets, CVIS emphasizes continual learning, enabling models to incorporate new information dynamically without retraining from scratch. Continual Video Instance Segmentation (CVIS) has the potential to significantly influence various aspects of society by enabling machines to dynamically learn and adapt from video data over time.

- Improved Automation and Efficiency: CVIS enhances automation in applications like autonomous vehicles, smart cities, and industrial monitoring, enabling real-time adaptation to changing environments and improving efficiency.
- Enhanced Public Safety: By enabling better anomaly detection and situational awareness in video surveillance and management, CVIS contributes to safer communities.
- Environmental and Wildlife Monitoring: CVIS can support long-term ecological studies, enabling the tracking of wildlife and monitoring of environmental changes without continuous human intervention.

Limitations: Although the proposed HVPL model can address the CIVS problem by alleviating catastrophic forgetting from both the image-level and video-level perspectives, it may struggle to continually learn large-scale semantically similar tasks. Therefore, we will explore how to increase the scalability of our proposed HVPL model in the future.