A Generalized Framework for Video Instance Segmentation

Appendix

A. Additional Implementation Details

A.1. Training

Freezing the frame-level detector [6] during training, we adopt pretrained weight from VITA [13] and finetune the video-level modules only; Object Encoder ($\mathcal{E}$) and Object Decoder ($\mathcal{D}$). Therefore, our total loss function is same as $\mathcal{L}_v$ describe in VITA and we employ same hyper-parameters. Concretely,

$$\mathcal{L}_v \text{ [13]} = \underbrace{\lambda_{cls}\mathcal{L}^v_{cls}}_{\text{categorical loss}} + \underbrace{\lambda_{ce}\mathcal{L}^v_{ce} + \lambda_{dice}\mathcal{L}^v_{dice}}_{\text{mask-related loss}}, \quad (4)$$

where the categorical loss is Cross Entropy and the mask-related loss consists of Binary Cross Entropy loss and Dice loss. The optimizer is AdamW, and we set the base learning rate and weight decay as 5e-5 and 5e-2 respectively, for all datasets. For YouTube-VIS 2019, we train for 15K iters and apply lr decay at 10K iters. For YouTube-VIS 2021/2022, we train for 90K iters and apply lr decay at 50K iters. And for OVIS, we train for 140K iters and apply lr decay at 100K iters.

A.2. Inference

We use the same inference procedure for all benchmarks and, once again, do not involve any heuristics. To further specify the inference procedure, we provide simplified PyTorch-style pseudo-codes of our GenVIS in Tab. 7. Given an input video, we sequentially process non-overlapping clips with predefined length N_f. For each clip, we obtain frame queries (`fq`) and mask features (`mf`) from Mask2Former [6] model by feeding backbone features. Then, the frame queries for the entire clip become the input of Object Encoder. After that, we generate clip-level predictions through Object Decoder by propagating instance queries (`q`) from previous clip with stacked memory (`memory`). Before going through the following clip, we add encoded instance prototypes (`p`) of the current clip to the memory.

B. More Experimental Details

We provide more experimental details about Tab. 4 in Sec. 4.4. For the matching algorithm of MinVIS [14] used in the baseline, we use the provided code from its official repository. Since there is no publicly available code for the **CL** (Correspondence Learning [32]), we reproduce the algorithm following the description from the original paper [32]. We apply the original target assignment algorithm of VITA on the first clip and keep the matched indices on the following clips. At the inference stage, we do not use heuristics of re-initialization of propagating queries for a fair comparison.

```
def GenVIS(video):
    pred_cls, pred_mask, memory = [], [], []
    q = object_decoder.q

    for clip in video:
        feats  = backbone(clip)
        fq, mf = mask2former(feats)

        fq     = object_encoder(fq)
        q, p   = object_decoder(fq, q, memory)

        memory.append(p)

        w = mask_head(q)

        # w.shape: (Nq x C)
        # mf.shape: (Nf x C x H x W)
        pred_mask.append(w @ mf)
        pred_cls.append(cls_head(q))

    return pred_cls, pred_mask
```

Table 7. PyTorch-style inference pseudo-code of GenVIS.

C. Additional Qualitative Results

In Fig. 5, we provide additional qualitative comparisons with state-of-the-art methods that provide checkpoints in official repositories. In addition to the results presented in Sec. 4.5, our method successfully captures highly occluded objects in long sequences.

D. Dependence on a large amount of labels

Designed from a frame-independent detector [6], MinVIS [14] has a great video-data efficiency. Despite using only 10% of the data, it still performs competitively, with -2.2 AP decrease compared to using all labels on OVIS [26]. On the contrary, the effectiveness of GenVIS is largely from its ability to model sequential characteristics of videos from consecutive clips. Thus, GenVIS does require more video-wise labels than MinVIS. Following the experimental settings of MinVIS, we trained GenVIS using only 10% of labels on OVIS with Swin-L. As can be expected, the reduction of video-wise labels hinders the temporal understanding: it results in drop of -9.7 AP ($45.2 \rightarrow 35.5$) compared to full labels. Our understanding is that GenVIS, which necessitates consecutive frames, faces challenges in learning diverse appearances compared to MinVIS, which uniformly samples frames, despite having the same limited number of frames.

E. Online inference speed

In Fig. 3, we analyze the trade-offs between performance and speed in online & semi-online setups. Thanks to its generalized property, GenVIS provides multiple options where users can consider either complexity of datasets or targeted runtime. Among the varying number of clip lengths, GenVIS shows 18.7 fps on the OVIS benchmark under the online setting ($N_f^{eval} = 1$). We measured the inference speeds of MinVIS [14] and IDOL [34] under a fair evaluation environment, and each achieves 28.1 and 2.1[3] fps, respectively.

F. Discussions

Accuracy with the increased number of frames. Current offline VIS methods [5, 13, 33] have difficulties in modeling sequential and temporal characteristics of long videos. Therefore, as GenVIS uses such methods for predictions within a window, the longer window brings about *inaccurate intra-clip predictions* for videos with complicated trajectories, leading to the performance drop (-12.4 AP) on the challenging OVIS benchmark (see Fig. 3 (b)). On the contrary, there is a marginal performance drop (-1.4 AP) when using longer windows on YouTube-VIS 2019, which comprises relatively simple trajectories. From the experiments, we would like to highlight our motivation that designing multiple inter-clip associations in a sequential manner is effective rather than simply enlarging a intra-clip window to handle challenging videos.

Accuracy gap between online and semi-online. Intuitively, as discussed in the above paragraph, semi-online methods with an adequate window size have more potentials to achieve higher accuracy than online models. However, compared to the semi-online versions of GenVIS, the online version can also achieve competitive accuracy as it stores more memories during evaluation. Affected by these characteristics, the optimal accuracy of GenVIS can be obtained in different settings as each backbone and dataset have different aspects. An interesting future direction would be to further enhance the intra-clip modeling, which would boost the performance of semi-online VIS, coupled with GenVIS.

Dataset licenses of COCO [21], YouTube-VIS [36], and OVIS [27]: Attribution 4.0 International, CC BY 4.0, and CC BY-NC-SA 4.0, respectively.

[3] IDOL shows the slow inference speed due to its increasing computations with respect to the number of frames for post-processing.

Figure 5. Qualitative comparisons of our method, GenVIS, with the state-of-the-art methods: MinVIS [14] and VITA [13]. GenVIS shows impressive accuracy in these complicated scenes where the objects look similar crossing each other. Objects with the same identity are displayed in the same color.