

VVitCutLER: Towards Unsupervised Object Detection and Segmentation in Videos

Supplementary Material

1. Other Annotation Method

In the main paper, we use additional methods referred to as VideoCut and CutSAM for comparison and analysis. For completeness and reproducibility, we provide their key details below.

1.1. VideoCut

VideoCut(see Fig. 1) is an unsupervised variant. Similar to our VitCut method, the first stage of VideoCut follows the VoteCut workflow (see the frame-level unsupervised target discovery section in the main paper). Therefore, we will not elaborate further here.

In the second stage, we first use the RAFT model [4] to estimate dense optical flow between consecutive frames, which excels at capturing fine-grained motion. Given a current frame I_t and a reference frame I_{t-1} , we compute forward optical flow $F_{t-1 \rightarrow t}$ and use it to warp the segmentation mask from the reference frame to the current frame. This generates a predicted mask M_t^{pred} based on temporal consistency.

Next, we compare the warped mask M_t^{pred} with the mask M_t^{curr} directly obtained from the first stage, computing the intersection over union (IoU) of their respective bounding boxes. If the IoU falls below a preset threshold, the object is considered unstable and discarded from the current frame. This filtering step helps eliminate temporally inconsistent transients or false detections.

In addition to motion-guided alignment, we also introduce a background recognition mechanism. For each masked region, we compute the average optical flow intensity. If the intensity is below a certain threshold across multiple consecutive frames, the region is labeled as `is_bg`, indicating that it likely belongs to the static background. We then group the masks across frames by their associated bounding box Intersection over Union (IoU) and track the frequency of the `is_bg` label. If a region is repeatedly labeled as background, we remove it from all subsequent frames.

1.2. CutSAM

CutSAM (see Fig. 2) is a fully supervised variant. Based on the bounding boxes produced in the first stage, we generate object masks using Segment Anything 2 (SAM2) [3]. The overall pipeline follows VideoCut, but with key modifications in the second stage to adapt to the supervised setting.

To enhance the reliability of these bounding boxes, we introduce a temporal consistency improvement process. Specifically, we compute the intersection over union (IoU)

of bounding boxes in the reference and target frames. When high IoU overlap is observed, the bounding box from the reference frame is merged into the bounding box of the target frame. This process reduces temporal noise and improves spatial alignment across frames.

Finally, the refined bounding boxes serve as hints to the Segmentation Model 2 (SAM2) [3] to generate updated high-quality masks. This step further refines the bounding boxes and segmentation masks, resulting in clearer and more accurate object representations.

2. Student Mask Generator and Distillation Details

2.1. Student architecture

The student mask generator predicts an instance mask and a confidence score from a cropped RoI. Given an RoI crop resized to 224×224 , we extract features using a frozen DINOv2-base backbone (dinov2-base). We take the last-layer patch tokens, remove the class token, and reshape them into a 16×16 feature map with $C=768$ channels. A lightweight decoder then outputs a mask of resolution 64×64 and a scalar confidence score.

Decoder configuration. We project the DINOv2 features with a 1×1 convolution to 256 channels, and apply a Transformer decoder with $L=6$ layers, $h=4$ attention heads, and feed-forward dimension 512. The decoder uses $K=16$ learnable queries (dimension 256) and a learned 2D positional embedding over the 16×16 grid. For mask prediction, we use a lightweight head consisting of two 3×3 Conv-BN-ReLU layers and a final 1×1 convolution, followed by bilinear upsampling to 64×64 . For confidence prediction, we apply global average pooling on the fused features and use a two-layer MLP to predict a scalar logit.

2.2. Teacher supervision

We use SAM2 as the teacher during offline distillation. Given a full image and an RoI box, the teacher predicts a binary mask and a confidence score. Teacher masks are aligned to the student output resolution (64×64) within the RoI region. After distillation, the teacher is not used during pseudo-label generation; we apply the distilled student checkpoint for efficient inference.

2.3. Loss

We follow the loss definition in the main paper (Eq.2 and Eq.3). The student is supervised by the teacher mask and

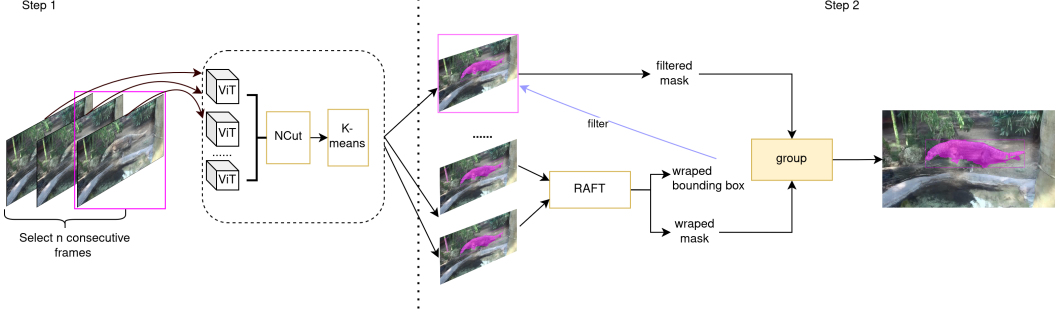


Figure 1. Overview of the proposed VideoCut framework for unsupervised mask extraction. Step 1: Multiple ViT model, NCut, and K-means are applied to each frame to produce initial segmentation masks. Step 2: RAFT is used to estimate dense optical flow between consecutive frames, which warps masks based on temporal motion. The warped masks are compared with the current frame’s masks using IoU, discarding inconsistent or transient objects. Additionally, a background recognition module analyzes average optical flow intensity to detect static regions, which are progressively removed to improve temporal stability.

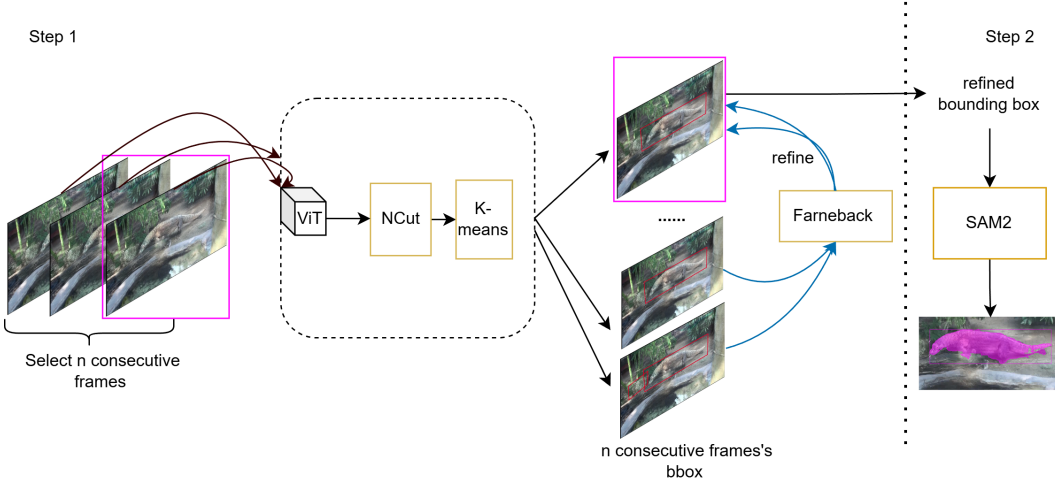


Figure 2. CutSAM’s overall architecture. The detected bounding boxes are further clustered to group related regions, and SAM2 is then applied to each cluster to predict a high-quality segmentation mask. This two-step pipeline achieves accurate object segmentation with minimal manual intervention.

confidence score, where $\mathcal{L}_{\text{score}}$ is binary cross-entropy on the confidence. The boundary term $\mathcal{L}_{\text{boundary}}$ is implemented by matching Sobel edge responses between the predicted mask probabilities and the teacher mask. The coefficients (0.5, 0.3, 0.2) are fixed across all experiments and are chosen to balance the relative contributions of the three terms, with BCE providing stable pixel-level supervision and Dice and boundary losses serving as complementary regularizers for region and contour refinement.

2.4. Training Details

We train the student decoder *from scratch* and freeze the DINOv2 backbone. Training is performed with distributed data parallelism. We use AdamW with learning rate 2×10^{-4} and weight decay 10^{-2} , batch size 64, for 40 epochs. We adopt a warmup cosine schedule with a restart: warmup

for 5 epochs, cosine decay until epoch 20, and a cosine restart for the remaining epochs, with minimum learning rate 10^{-6} . We apply simple data augmentation with random horizontal/vertical flips (each with probability 0.5).

Teacher-score filtering. To reduce noisy supervision, we discard RoIs with low teacher confidence during distillation. Specifically, we keep only samples with teacher score > 0.7 within each batch.

3. Qualitative Analysis

In this section, we present further qualitative results to demonstrate the quality and consistency of VitCut used in this study. We include examples from the datasets described in the main paper, as well as supplementary datasets collected for expanded evaluation. These visualizations high-

light the accuracy of the annotation process, the diversity of scenarios, and the robustness of our annotation protocol.

3.1. Additional Datasets

YouTube-VIS 2019 contains 2,238 videos over 40 categories and serves as a standard benchmark for video instance segmentation under realistic motion and appearance variations.

OVIS includes 901 videos across 25 categories and focuses on severe occlusions and crowded scenes, making it a demanding benchmark for occlusion-robust VIS.

3.2. Qualitative Results of VitCut

The qualitative comparisons in Figs. 3 and 4 highlight the strengths and limitations of different annotation generation methods. Compared to VideoCut, CutSAM, and VoteCut, our method (**VitCut**) generates pseudo masks that better match the ground truth and exhibits higher temporal stability across consecutive frames. These results indicate that our Transformer-based design effectively captures long-term dependencies and accommodates changes in object appearance over time.

Nevertheless, a noticeable gap remains between VitCut and the supervised CutSAM baseline. In high-resolution examples, the boundaries produced by VitCut can be less smooth and may miss fine contour details. This is likely due to the combination of (i) the lightweight decoder design, which prioritizes efficiency, and (ii) the limited granularity and noise in the pseudo-annotation training signal, making it challenging to recover highly detailed object contours.

Among the baselines, VideoCut shows the most significant degradation. Its mask quality is strongly affected by optical flow errors, especially under strong motion blur, occlusion, rapid displacement, or non-rigid deformation. In these scenarios, distorted masks can propagate incorrect information across frames, leading to shape drift or severe distortion. This suggests that optical flow-based cues are better used as supplementary signals rather than a fully reliable basis for mask propagation.

Despite these limitations, the results on additional datasets in Fig. 4 demonstrate that VitCut generalizes well to benchmarks such as YouTube-VIS 2019 and OVIS. Even without training on these datasets, the method consistently generates stable and coherent instance masks, highlighting the robustness of the proposed annotation pipeline.

3.3. Qualitative Results of VVitCutLER

Beyond generating high-quality pseudo masks for individual instances, the complete **VVitCutLER** pipeline can handle complex multi-object scenes. As shown in Fig. 5, the framework can identify multiple objects and maintain mask consistency over time, even under notable changes in appearance and motion.

Annotation Type	Model	R1 mAP ₅₀ ↑		R2 mAP ₅₀ ↑		AR(L) ↑		mFPS ↑
		BBox	Segm	BBox	Segm	R1	R2	
Unsupervised	CutLER[5]	28.43	-	32.46	-	21.1	26.5	16.68
	CutvLER[1]	28.95	-	33.96	-	23.2	29.6	16.1
	VVitCutLER	38.26	-	43.68	-	37.5	43.7	14.93
	vs CutLER	+9.83	-	+11.22	-	+16.4	+17.2	-1.75

Table 1. **ImageNetVID results under the pseudo-label (unsupervised) setting across Round 1 (R1) and Round 2 (R2).** We report mAP₅₀ and AR(L) for each round, and mFPS for efficiency. Since ImageNetVID is detection-only, the segmentation columns are not applicable (shown as “-”). Green numbers denote absolute gains over the CutLER baseline.

Annotation Type	Model	R1 mAP ₅₀ ↑		R2 mAP ₅₀ ↑		AR(L) ↑		mFPS ↑
		BBox	Segm	BBox	Segm	R1	R2	
Unsupervised	CutLER	35.21	28.54	39.04	29.03	28.2	33.8	18.98
	CutvLER	36.71	29.01	39.94	30.93	29.2	34.2	19.1
	VVitCutLER	37.02	31.43	41.12	32.48	35.2	39.3	14.32
	vs CutLER	+1.81	+2.89	+2.08	+3.45	+7.0	+5.5	-4.66

Table 2. **YouTube-VIS results under the pseudo-label (unsupervised) setting across Round 1 (R1) and Round 2 (R2).** We report both bounding-box and segmentation mAP₅₀, as well as AR(L) for each round, and mFPS for efficiency. Green numbers denote absolute gains over the CutLER baseline.

However, since the pipeline is entirely unsupervised, certain challenging cases remain. When objects move very close to each other or share similar visual patterns, the model may confuse identities or merge them into a single instance. This limitation mainly stems from the absence of explicit supervision signals, which makes it difficult to robustly separate instances in highly cluttered or interactive scenes.

4. decoder performance

To evaluate downstream performance under pseudo-label supervision, we report results on ImageNet-VID and YouTube-VIS across two rounds of self-training (R1/R2). All experiments are conducted in a CNN detector architecture: we adopt Mask R-CNN with a ResNet-50+FPN backbone, and incorporate temporal information only at the RoI-feature level via SELSA while keeping the prediction heads unchanged. We compare VVitCutLER against published unsupervised baselines, including CutLER [5] and CutvLER [1], under the same evaluation protocol.

As shown in Tabs. 1 and 2, VVitCutLER consistently improves over prior baselines on both datasets and in both self-training rounds. On ImageNet-VID (detection-only), VVitCutLER increases mAP₅₀ from 28.43/32.46 (CutLER) to 38.26/43.68 in R1/R2, and yields substantial improvements in AR(L) (+16.4 and +17.2 for R1 and R2, respectively). On YouTube-VIS, VVitCutLER improves both bounding-box and segmentation accuracy, reaching 37.02/31.43 (BBox/Segm) in R1 and 41.12/32.48 in R2, while also achieving higher AR(L) in both rounds. The reduction in mFPS reflects the expected accuracy-speed trade-off introduced by temporal RoI feature aggregation.

DAVIS

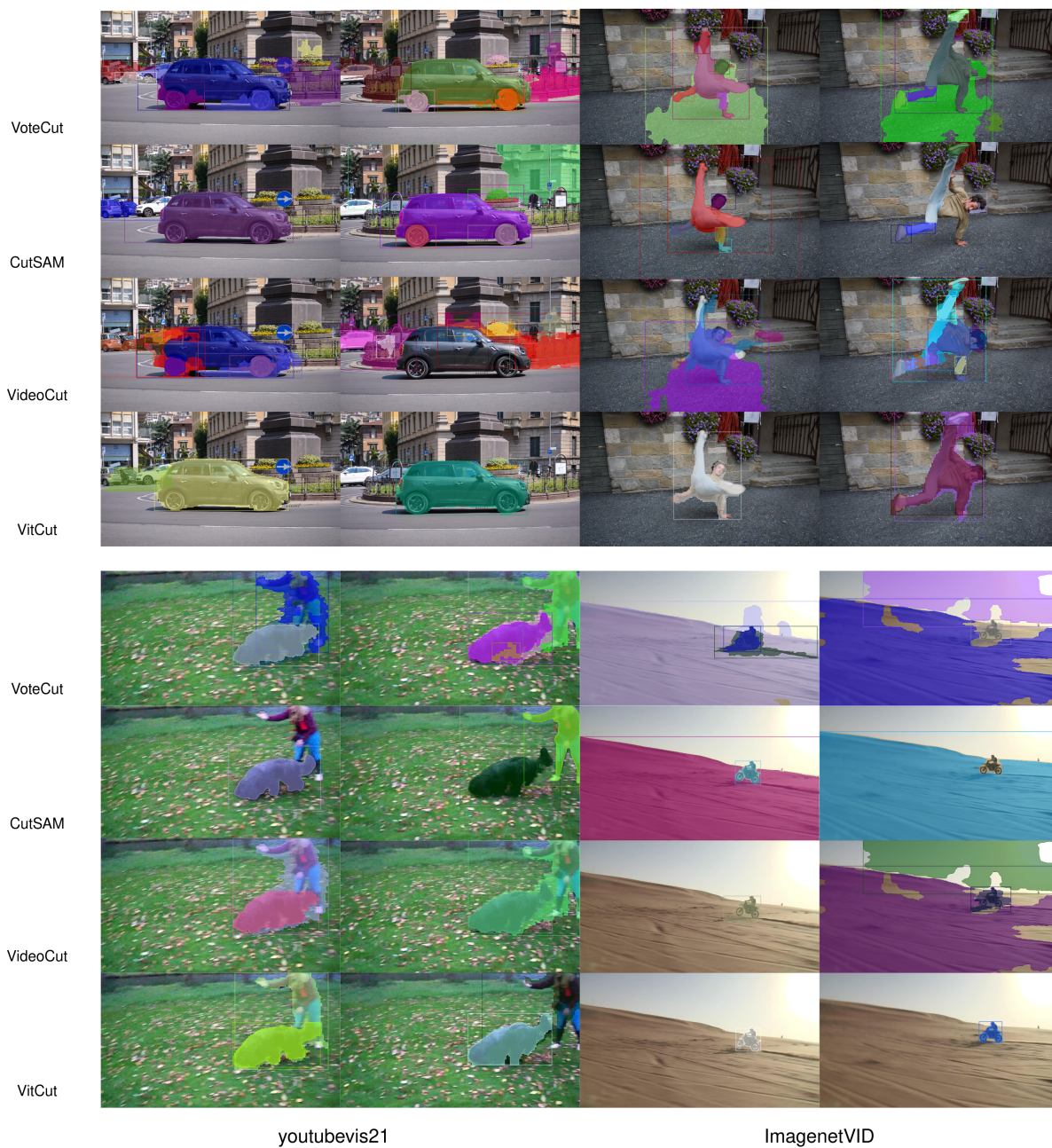


Figure 3. Qualitative visualizations on YouTube-VIS 2021, DAVIS, and ImageNet-VID.

Overall, these results demonstrate that VVitCutLER provides stronger downstream detection and segmentation performance under unsupervised self-training.

5. Ablation

5.1. Feature Selection

To analyze the impact of candidate box filtering on our detector, we evaluated different Top-K settings for selecting RPN candidate boxes. Instead of using a fixed confidence threshold, we retained the top K bounding box predictions sorted by RPN confidence score. We experimented with

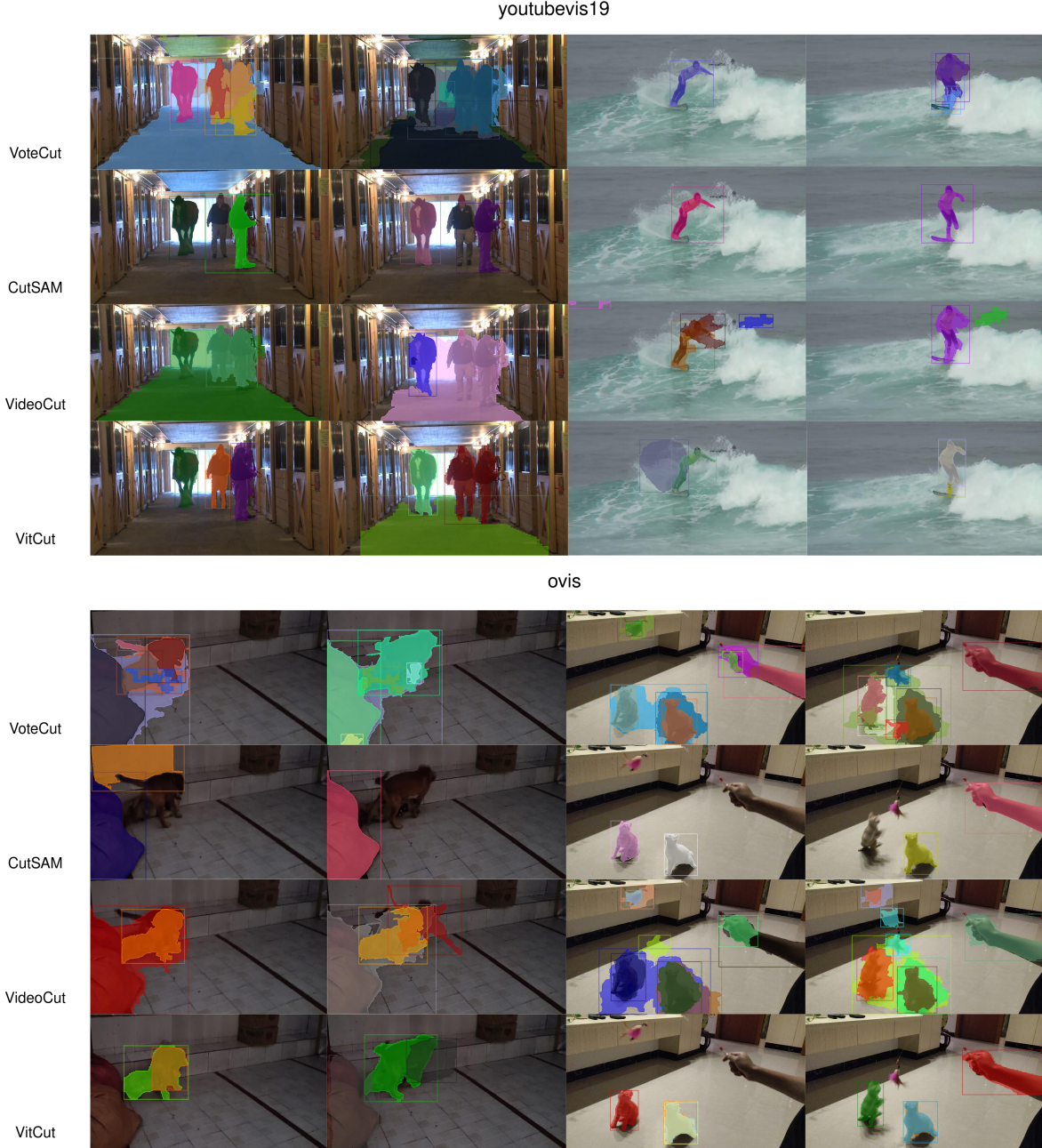


Figure 4. **Qualitative visualizations on YouTube-VIS 2019 and OVIS.** We compare different annotation generation methods on two additional video instance segmentation benchmarks that are not used for training. The results show that **VitCut** produces more stable and refined masks across diverse scenarios, indicating strong cross-dataset generalization.

$K \in \{30, 100, 120, 150, 200\}$ and calculated the average recall (AR) on the training set.

As shown in Fig. 6, while increasing the K value consistently improves the AR value, indicating that retaining more candidate boxes allows the detector to capture more potential foreground regions, the runtime also increases al-

most linearly. This is because a larger K value requires subsequent stages (including feature extraction, classification, and mask prediction) to process more candidate boxes, directly increasing the computational cost per frame.

From an efficiency perspective, Top-K = 150 achieves the best balance between accuracy and cost, achieving an



Figure 5. Qualitative visualizations of VVitCutLER on YouTube-VIS 2021 and ImageNet-VID.

AR value of 96.18% within a 100ms runtime per frame. While increasing the K value to 200 provides a 0.47% improvement in AR value, the runtime increases by 50%, resulting in diminishing returns. For video applications, maintaining low frame latency is crucial, making this additional overhead unacceptable.

In conclusion, selecting the top 150 detector candidates achieves an effective balance between recall and efficiency and serves as the default configuration for our system.

5.2. Aggregation on Cascade R-CNN

To evaluate whether our aggregation strategy can be generalized to more complex detector architectures, we further experimented on Cascade R-CNN[2], a multi-stage detection framework with iterative feature refinement. We inte-

grated the aggregator module at different stages and conducted evaluations.

As shown in Fig. 7, none of the fusion strategies improved upon the unmodified Cascade R-CNN baseline model. The baseline model still achieved the best overall performance (16.62% bounding box mAP and 12.33% segmentation mAP), while the insertion aggregator typically reduced accuracy—especially in stage 3 and all-stage settings.

This phenomenon can be attributed to the multi-stage refinement mechanism of Cascade R-CNN. Each stage relies on clean and progressively refined features; therefore, injecting temporally aggregated features that may contain noise or redundancy can disrupt the refinement process, leading to suboptimal predictions. In contrast, simpler

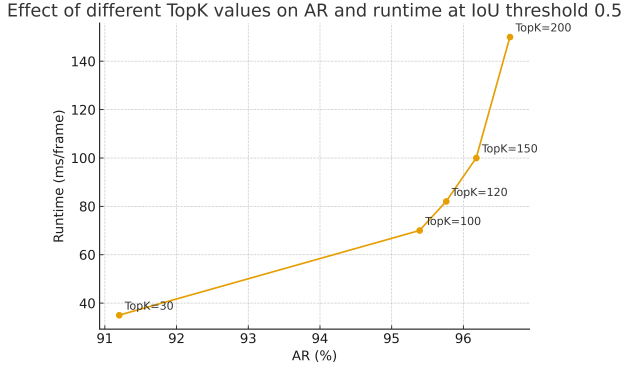


Figure 6. **Effect of different TopK values on AR and runtime at IoU threshold 0.5.** TopK=150 achieves the best balance between accuracy and efficiency.

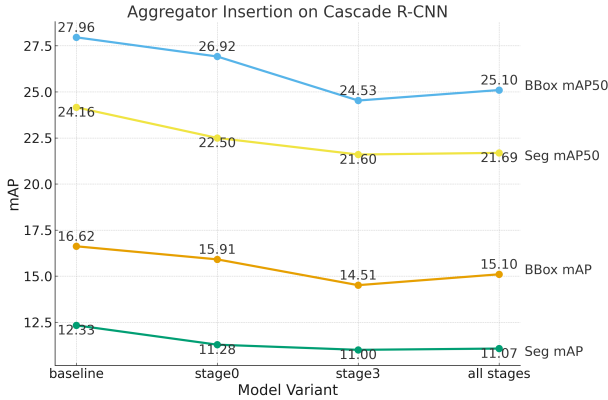


Figure 7. **Effect of inserting the aggregator module into different stages of Cascade R-CNN on YouTube-VIS.** The baseline achieves the best performance, while inserting the aggregator at various stages results in performance degradation.

single-stage or two-stage detectors are more tolerant of temporal fusion and can benefit from our aggregation design.

5.3. Effect of Teacher Model Choice

To explore how different teacher models affect the quality of pseudo-labels, we compared three candidate models: DINOv2-base, DINOv2-large, and SAM2.

We evaluated each teacher model using mean intersection-over-union (mIoU) and rating accuracy, both of which reflect the quality and reliability of pseudo-labels. As shown in Tab. 3, SAM2 achieved the highest mIoU value (0.6889), approximately 3.7% higher than DINOv2-base and significantly better than DINOv2-large (0.3183). This indicates that SAM2 generates more accurate and finer-grained segmentation masks, providing higher-quality supervision information.

While all teacher models exhibited high score accuracy, SAM2 achieved a perfect score of 1.0000, demonstrating

Teacher Category	Teacher Model	mIoU \uparrow	Score Acc. \uparrow
Teacher Models	DINOv2-base	0.6644	0.9973
	DINOv2-large	0.3183	0.9988
	SAM2	0.6889	1.0000

Table 3. **Comparison of pseudo-label quality produced by different teacher models.** SAM2 achieves the highest segmentation quality (mIoU) and perfect score accuracy, demonstrating its advantages as a teacher model for pseudo-label generation.

stronger stability and consistency. DINOv2-large achieved slightly higher score accuracy than DINOv2-base, but its segmentation quality dropped sharply, making it unsuitable for pseudo-label generation.

Overall, SAM2 strikes the best balance between segmentation accuracy and label reliability. Its segmentation-oriented design enables it to generate fine-grained, high-quality masks, making it the most effective teacher model in our student training pipeline.

References

- [1] Shahaf Arica, Or Rubin, Sapir Gershov, and Shlomi Laufer. Cuvler: Enhanced unsupervised object discoveries through exhaustive self-supervised transformers, 2024. 3
- [2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection, 2017. 6
- [3] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. 1
- [4] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. 1
- [5] Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation, 2023. 3