Disentangling spatio-temporal knowledge for weakly supervised object detection and segmentation in surgical video- Supplementary Materials

A. Implementation Details

A.1. Teacher student module details

The teacher and student networks have four layers each. The key difference is that each layer of the teacher network uses an MLP, while the student network uses 3D-CNNs for spatio-temporal convolution. A unique feature of the teacher network is that, to help the backbone learn generalized features at different scales, features from each MLP layer are randomly downsampled during training with a probability of 0.5 before the ranked spatial and temporal pooling. This could be considered a heuristic learning strategy that is close to approaches of randomly masking out feature patches [13,28]. More details about the teacher network's forward operation are shown in the pseudo code in Table S1. As the student uses 3D-CNN instead of MLP, while keeping the overall architecture the same, student's network size and computational load are higher than the teacher's, as presented in Table S2. Link to our code: https://github.com/PCASOlab/VDST-net.

A.2. Training details

We implemented our training on an Nvidia A5500-based compute platform. Both the teacher and student networks contain a dropout layer after either the MLP (in the teacher) or the 3D-CNN (in the student). For the first nine epochs, only the teacher network was optimized with a dropout rate of 0.5. Subsequently, the student module was enabled with a dropout rate of 0.5, while the teacher's dropout was disabled. After this, both networks were trained for 30 epochs. Optionally, either the teacher or the student can optimize the backbone. For instance, if the teacher is allowed to fine-tune the backbone, the gradient flow from the student to the backbone is stopped.

B. Transient Object Presence Cholec80 Data Statistics

Details of the complete Cholec80 video clips set and the sampled training set are presented in Table S3. Certain instruments, like the Hook, have a high presence frequency throughout the overall dataset or within video. The latter is quantified by the percentage of frames per video (FPV). Instruments such as Scissor and Bipolar appear less frequently, with FPV percentages around or below 60%.

Table S1. Numpy-like Pseudo-code for teacher branch forward.

```
# image_encoder - ResNet or Vision Transformer
# W_MLP [c_in, c_e]- one layer of proj of the MLP
# W_fc [c_e, N] - learned proj of feature to CAM/Class
# v[bz, d, h, w ,c] - minibatch of videos
# labels[bz, N] - minibatch of ground truth class

# extract feature representations of video
T_f = image_encoder(v) #[bz, d, h_f, w_f, c_f]
T_e = T_f.reshape[bz* d* h_f* w_f, c_f]

for W_MLP in MLP_layers:
    # One layer of MLP
    T_e = ReLU(np.dot(T_e, W_MLP), axis=1)
    #  prob 0.5 of downsample
    if Random(0,1) > 0.5:
       T_e.reshape[bz, d, h_new, w_new, c_e]
       # avg pool with kernel k, and stride s
       T_e = Avg_pool_3D (T_e,k=[1,2,2],s=[1,2,2])
       T_e.reshape[bz* d* h_new* w_new* c_e]

T_e.reshape[bz, d, h_new, w_new, c_e]
# intepolate in spatial
T_e = 3D_interpolate (T_e) #[bz, d, h_f, w_f, c_e]
#ranked spatial pooling
slice_valid = Rank_pool_hw(T_e,k1) # [bz,d,1,1,c_e]
#ranked temporal pooling
final_f = Rank_pool_d(slice_valid,k2) # [bz,1,1,1,c_e]
final_f.reshape(bz,c_e)

#loss for training
class_logit = sigmoid (np.dot(final_f, W_fc),axis=1)
loss_t = BCE_loss(class_logit, labels, axis=1)
#ST_CAM interface, N class activation maps
T_e.reshape[bz* d* h* w, c_e]
ST_CAM = RelU(np.dot(T_e, W_fc),axis=1)
ST_CAM.reshape[bz, d, h_f, w_f, N]
```

Table S2. Teacher and student parameter overview with video feature input. FLOPs: Floating-point operations per second.

Parameter	teacher	student
Size	4.02 MB	17.02 MB
Trainable params	1.05 M	4.46 M
Forward/backward pass size	1325.39 MB	2092.8 MB
FLOPs	31.48 G	132.68 G

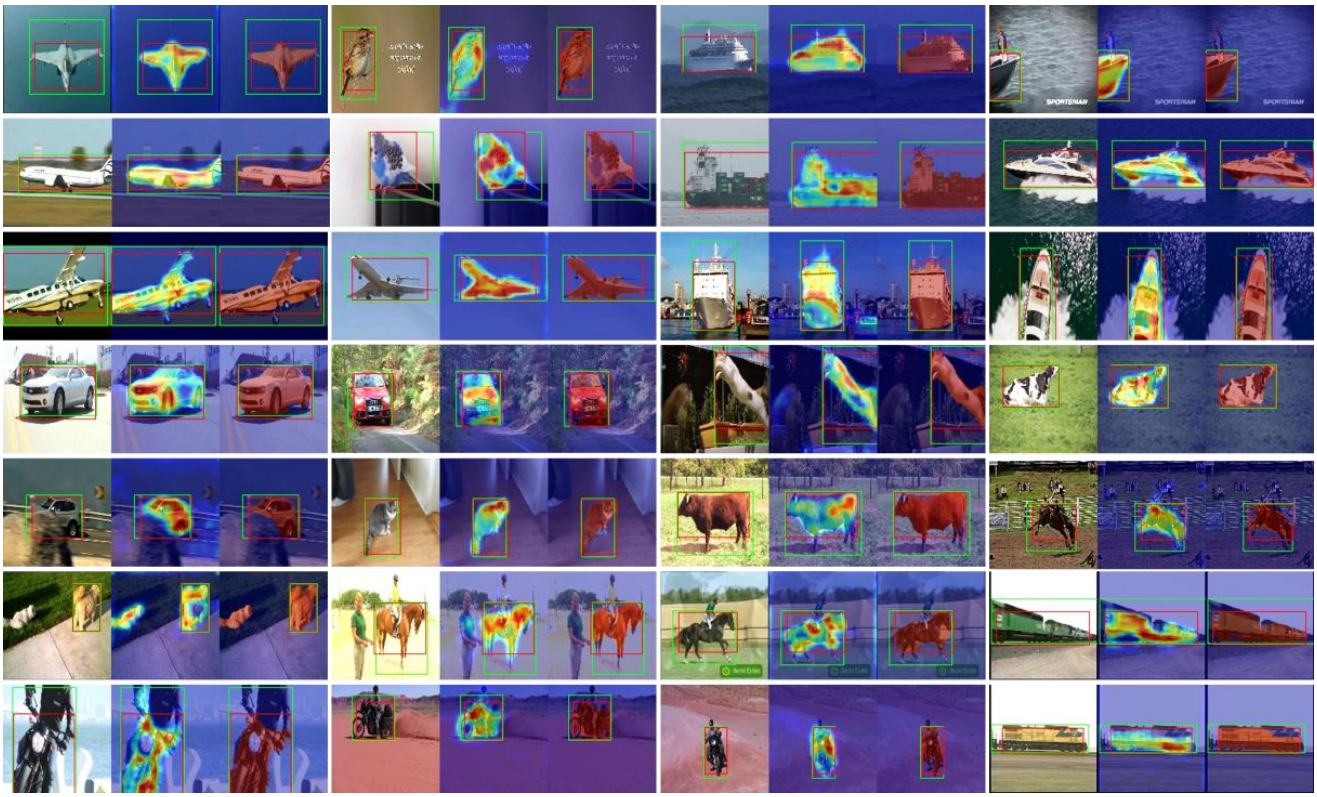

Figure S1. More qualitative results of the proposed method on Youtube-objects data. Original images, activation maps, and masks after thresholding are presented sequentially. Our method demonstrates robustness, producing activation maps and segmentation masks that accurately follow the objects' contours, with minimal false activation on the background.

Table S3. Statistics of the Transient Object Presence (TOP) Cholec80 Data. CholecSeg8K removal: Removal of overlapping clips annotated by CholecSeg8K from the original Cholec80 dataset, as described in section 4.1.

	Cholec80		CholecSeg8K removal		
Category	Frames	Clips	Frames	Clips	FPV
Grasper	102k	5k	83k	4k	63.6%
Bipolar	9k	0.59k	7k	0.48k	48.2%
Hook	103k	4k	87k	3k	80.8%
Scissor	3k	0.24k	3k	0.19k	46.6%
Clipper	6k	0.35k	5k	0.28k	57.0%
Irrigator	10k	0.62k	8k	0.51k	52.7%
Specimen bag	11k	0.62k	9k	0.51k	60.7%

Table S4. Results of using separate backbones for teacher and student respectively, CorLoc score on the Youtube-Objects data are reported.

	Full	T only
Res teacher + ViT student	**75.0**	68.5
ViT teacher + Res student	68.4	63.3

Table S5. Results of using Resnet34 as feature extractor on Cholec80 data.

	IoU[%] ↑	Dice[%]↑	HD[pix]↓
VDST-Net	43.58	46.15	53.50

C. Additional Results

We also present results using a ViT backbone for the student and a ResNet backbone for the teacher, as well as the reverse configuration. As shown in Table S4, both configurations improve performance over the teacher alone. Notably, using ViT as the student results in the highest performance among them.

Additionally, we utilize a ResNet-34 backbone for the TOP cholec80 videos (Table S5). While this approach shows a decline in accuracy compared to our ViT-based method, it still achieves comparable performance to other state-of-the-art methods, as presented in Table 2.

More qualitative results on the YouTube-Objects dataset are presented in Figure S1.