End-to-End Semi-Supervised Learning for Video Action Detection (Supplementary)

Akash Kumar Yogesh Singh Rawat
Center for Research in Computer Vision
University of Central Florida
akash_k@knights.ucf.edu yogesh@crcv.ucf.edu

- Section 1: Dataset details used in our experiments.
- Section 2: Training details for ablation studies.
- Section 3: Training details for baseline semi-supervised approaches extended to videos domain.
- Section 4: Some additional discussions
- Section 5: Qualitative analysis of samples

1. Dataset

K400 [5]: Kinetics-400 contain youtube videos. For our work, we conduct experiments on 1%, 2% and 3% of the total dataset size. The total number of videos are 300K, out of which 240K videos are for training. We pick unlabeled subset from these videos randomly.

UCF101 (77) [6]: This dataset includes 77 activities of UCF101 dataset which are not present in UCF101-24 action detection datasets. The total number of videos for training include around 7k. We use split-1 of UCF101 dataset.

2. Training Details for main algorithm and ablations

2.1. Action Detection

Untrimmed: For training on untrimmed dataset, instead of picking the annotated frames, the frames are picked at random. Composition of labeled vs unlabeled subset is 20/80 which is similar to trimmed dataset experiments.

Additional loss for localization: In addition to the supervised localization loss (BCEwithLogits), we incorporate Dice loss [7] in our training. The reason behind choosing Dice loss along with BCE with Logits loss is that it focuses on class imbalance, it readjusts the pixel weights. Dice loss maximizes the intersection between prediction and ground truth, and at the same time minimizes the union. The dice coefficient is equivalent to the F1 score. We take harmonic mean over union and intersection. The harmonic mean is always biased to give a lower score. Thus, the Dice coefficient is more suitable. We can see the benefit of using Dice loss alongwith BCE with logits in Table 1. There's an absolute gain of roughly 2% at 0.5 mAP. The scores are average of three runs.

Losses	f-mAP (%)		v-mAP (%)	
	0.2	0.5	0.2	0.5
BCE	88.2	67.9	94.5	70.0
BCE+Dice	89.6	69.8	95.2	71.8

Table 1. Effect of using Dice loss

Epoch	f-mAP (%)		v-mAP (%)	
	0.2	0.5	0.2	0.5
5	89.6	69.2	95.6	72.0
10	90.0	69.9	95.7	72.1
15	88.1	67.8	94.2	69.7

Table 2. Effect of introducing pseudo labels at different epochs

Frames	f-mAP (%)		v-mAP (%)	
	0.2	0.5	0.2	0.5
3	89.1	69.8	95.3	71.9
5	90.0	69.9	95.7	72.1

Table 3. Using more number of frames

Selection of number of frames for variance calculation: For the calculation of variance, we take 5 frames into consideration to calculate variance for single frame. We compare the performance of 5 frames vs 3 frames. Table 3 shows that calculation of variation over 5 frames outperformed the variance using 3 frames proving that longer temporal aspect does help. We see a jump in performance especially at 0.2 metrics for both f-mAP and v-mAP.

Effect of introducing pseudo-labels at different

stages: VideoCapsuleNet was proposed for supervised learning where the video level class labels are used for capsule masking during localization. The masked capsules are then upsampled multiple time to get the localization mask. The masking of capsules is optional and attaching a class label to the bounding box helps the network to relate attributes for that particular class. In our work, a single batch contains labeled and unlabeled data. For labeled data, we use the ground truth as in supervised. In case of the unlabeled data, for first few epochs, we do not attach any label to the bounding box. It's because the network is not generalized on the training dataset. After a couple of epochs, the network is fine-tuned on the target dataset, then we predict pseudo-labels for the unlabeled samples in the batch. Class prediction is multiplied with the poses and then upsampled to get the localization map. As the prediction becomes more confident and accurate with the training, the prediction improves with time. We pick epoch 10 in our training protocol to start introducing pseudo labels for the unlabeled samples. (Table 2) If we start predicting pseudo labels too early, then, we see a slight decrease in performance. Introducing pseudo labels at 15 makes it susceptible towards dataset imbalance.

2.2. Video object segmentation

For LSTM approach, we consider 32 sequential frames from a video. To get the intermediate frames, between provided annotated frames from the original dataset, we perform interpolation. We train the network for 50 epochs.

3. Training Details for semi-supervised baselines

For all of the baseline semi-supervised approaches, on top of supervised classification loss, we did measure the spatio-temporal localization loss proposed in our original work, i.e., BCE with Dice loss on labeled samples.

MixMatch [1]: We generated two views one with a weak augmentation and one with a string augmentation. For weak augmentation, we just flip the original view. For strong augmentation, a series of augmentation is randomly applied such as multiscale cropping and random flipping, etc. There's a limitation of directly extending image approaches. The batch size of each labeled and unlabeled dataset is eight.

Pseudo-label [3]: To get the results on pseudo-label, we did three iterations of training. First iteration is on the baseline subset that is 20% of the dataset. Next, we generate pseudo labels for the whole data, we pick the top 10% based on confidence threshold. Remove the ones that overlaps with the 20% labeled subset. Add remaining pseudo labels as a training dataset in the next iteration. Then, finetune on the dataset utilizing the pretrained weights of previous iteration. We perform three iterations and then evaluated to get the final result.

Co-SSD(CC) [4]: This procedure is similar to training with our classification consistency standalone training.

Unlabeled	f-mAP (%)		v-mAP (%)	
	0.2	0.5	0.2	0.5
1x	86.7	64.8	93.5	65.6
2x	87.8	67.5	94.4	69.6
3x	88.3	69.3	94.7	71.0
4x	89.6	69.8	95.2	71.8
Full	89.6	69.8	95.2	71.8

Table 4. Varying the amount of unlabeled data from 20% to 80%. Labeled data is kept constant at 20%.

Weights	UCF101-24		JHMDB-21	
	f-mAP	v-mAP	f-mAP	v-mAP
Pre-trained	69.9	72.1	64.4	63.5
w/o Pre-trained	60.8	58.7	34.3	29.0

Table 5. Effect of using Pretrained weights. Results for 0.5 mAP. w/o - without

4. Discussions

Pretrained weights We assess the effect of utilizing pretrained weights in boosting performance. All of the previous approaches use 2-D or 3-D networks and are pretrained on some dataset. 2-D networks are trained on ImageNet and COCO dataset, whereas, most of the 3-D networks are pretrained on Kinetics-400 [5] and Kinetics-600 [2]. Kinetics datasets contains actions that incorporates sports actions such as *playing basketball*, *playing cricket*, *playing tennis* and more. From Table 5, we see that at 0.5 mAP, there's a roughly 10% decrease in performance. For JHMDB-21, the performance drop is by a significant margin of 30%. This shows the importance of utilizing pretrained weights especially for small scale datasets such as JHMDB-21.

Unlabeled Dataset: In table 4, we include performance on 0.2 metrics alongwith 0.5 extending the table present in original paper.

Data percentage (multiple seed) Multiple runs for different percentage of dataset. In the main paper, we show the improvement with increase in the amount of labeled videos is for only one seed variation. However, if the dataset is small then there's more variation in performance. We perform three different seed runs and show the results with variance calculation in Table 7.

Experiment	f-mAP		v-mAP	
	0.2	0.5	0.2	0.5
Variance	89.2 ± 1.70	61.9 ± 1.90	94.1 ± 0.60	61.4 ± 1.35
Cyclic Variance	88.6 ± 1.90	63.0 ± 1.30	94.1 ± 0.80	61.5 ± 0.90
Variance + L2	87.9 ± 1.55	63.3 ± 1.01	94.6 ± 1.09	62.4 ± 1.05
Cyclic Variance + L2	**89.9** ± 2.10	**64.4** ± 1.25	**95.4** ± 1.10	**63.5** ± 0.65
Gradient	88.1 ± 1.25	63.2 ± 1.70	95.3 ± 0.35	63.1 ± 2.55
Gradient + L2	88.0 ± 1.30	63.1 ± 1.95	94.4 ± 0.90	62.2 ± 2.45

Table 6. An analysis on temporal constraints for consistency regularization using JHMDB-30% dataset. The gain is absolute to the base case where only non-weighted L2 spatio-temporal consistency is utilized.

Composition	f-mAP@0.5	v-mAP@0.5
5/95	58.8 ± 1.30	57.6 ± 1.75
8/92	63.5± 0.60	64.3 ± 0.40
10/90	65.1 ± 0.55	66.2 ± 0.70
15/85	67.7 ± 0.40	69.6 ± 0.80
20/80	69.3 ± 0.40	71.1 ± 0.10

Table 7. Seed variation (three runs) for different subset of data.

Augmentation	f-mAP (%)		v-mAP (%)	
	0.2	0.5	0.2	0.5
HF	89.6	69.8	95.2	71.8
HF+MC	89.6	68.8	94.8	71.6

Table 8. Effect of using augmentations on UCF101-24. HF - Horizontal flipping, MC - Multi-cropping.

Unlabeled	f-mAP (%)		v-mAP (%)	
Dataset	0.2	0.5	0.2	0.5
Sup. (100)	89.4	69.2	95.3	71.9
UCF101(77)	91.7	74.8	96.5	78.1
K400 (1%)	89.7	71.2	95.0	72.6
K400 (2%)	90.5	73.1	96.4	76.2
K400 (3%)	91.0	73.8	96.8	75.8

Table 9. Use of extra data as a supervisory signal. Experiments on UCF101-24.

Ablation study (multiple seed) Different seed runs for ablations. The main paper discusses the mean score for JHMDB dataset. Since, it's a small dataset and there's an issue of overfitting, here we include the table with variance for JHMDB. (Table 6)

Spatial Augmentations: In addition to horizontal flipping, we run experiments with some strong spatial to see how it impacts classification consistency and spatio-temporal consistency. To recognize the impact of augmentations, we pick the baseline spatio-temporal consistency model. In addition to horizontal flip, we include multi-scale cropping with ratio varying from 0.7 to 1.0. From Table 8, the scores are consistent except for f-mAP@0.5. We see that there's a drop by 1% at f-mAP@0.5.

Bigger subset of Kinetics: We did more experiments of even bigger subsets of kinetics with 3 percent as unlabeled dataset. We want to see how much large-scale we can go and how much it impacts the gain in performance or does it saturates after a point. From table 9, we see incorporating 3% of kinetics dataset reflects some gains for f-mAP, but, there's a performance drop for v-mAP.

5. Qualitative Analysis

Here, we include the results for multiple samples from UCF101-24 datasets. Especially, we compare the output of different semi-supervised approaches that served as baselines in the paper. There are six samples in Figure 1 to show the robustness of proposed approach. The samples are a sequence of eight frames. The bounding box is shown if the predicted localization has an overlap greater than or equal to 0.5 with the ground truth localization map. If we look into first sample, pseudo-label fails to cover the whole actor and classification consistency captures more area than the ground truth. In sample 2 and 3, we can see that our approach works better even for small objects present in the scene. The actions are also rapidly changing across frames in this video. In both the samples, our approach is able to localize the actor with very high precision in comparison to other two semi-supervised approaches. For sample 4 outputs, Co-SSD is capturing the rocks even though actor area is dominant. Our results are very close to the ground truth. From sample 5, we want to convey that when the actor is

stable and performing action at the same position, then, all the three approaches correctly detects the location of actor. In sample 6, Co-SSD and pseudo-label have a lot of mis-predictions as compared to our approach. That shows the robustness of using variance across frames.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[2] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *ArXiv*, abs/1808.01340, 2018. 2

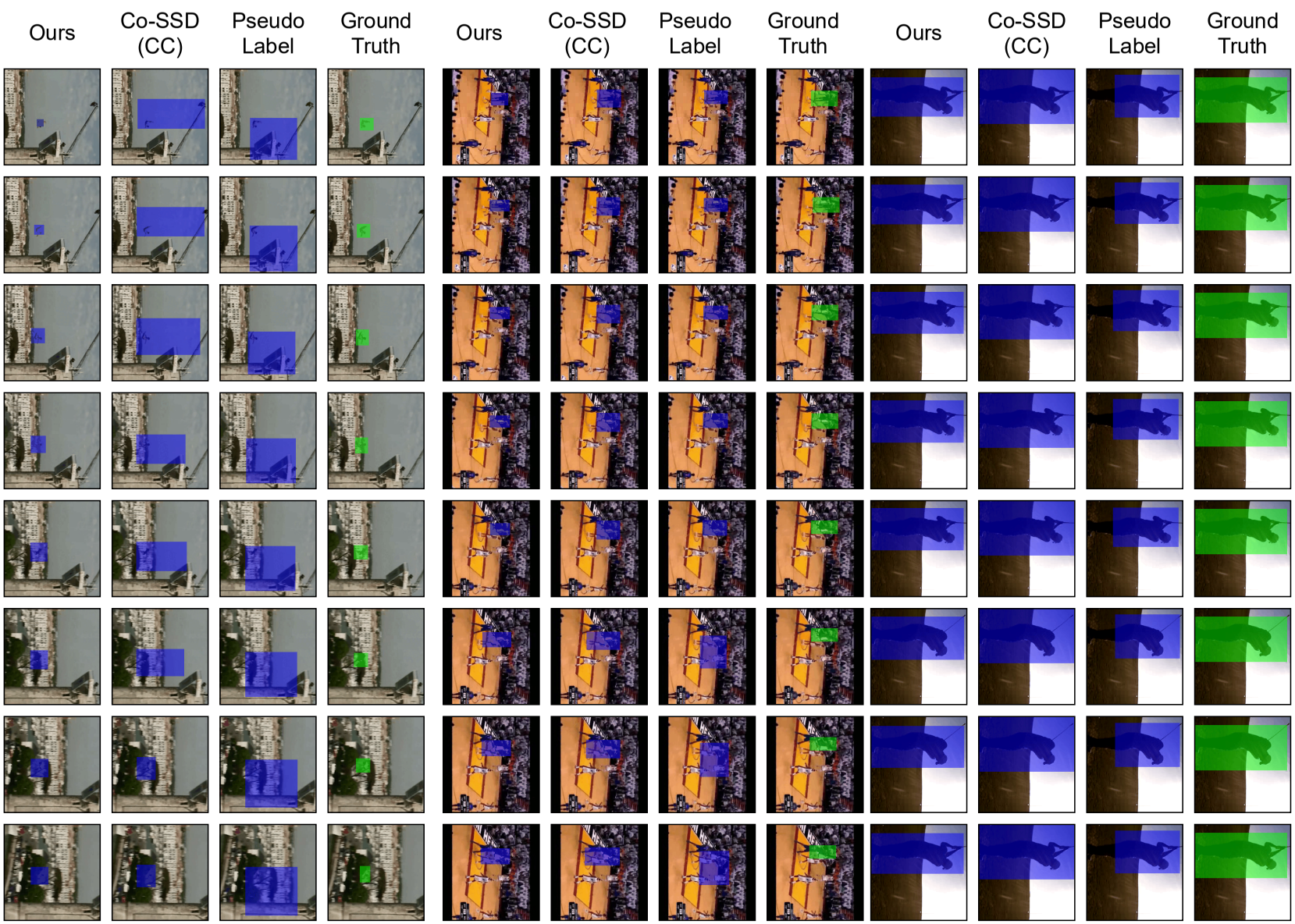
[3] Dong hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. 2, 6

[4] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2, 6

[5] Will Kay, João Carreira, K. Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, T. Back, A. Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 1, 2

[6] K. Soomro, A. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012. 1

[7] Carole H. Sudre, Wenqi Li, Tom Kamiel Magda Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,...*, 2017:240–248, 2017. 1


Ours
Co-SSD (CC)
Pseudo Label
Ground Truth
Ours
Co-SSD (CC)
Pseudo Label
Ground Truth
Ours
Co-SSD (CC)
Pseudo Label
Ground Truth

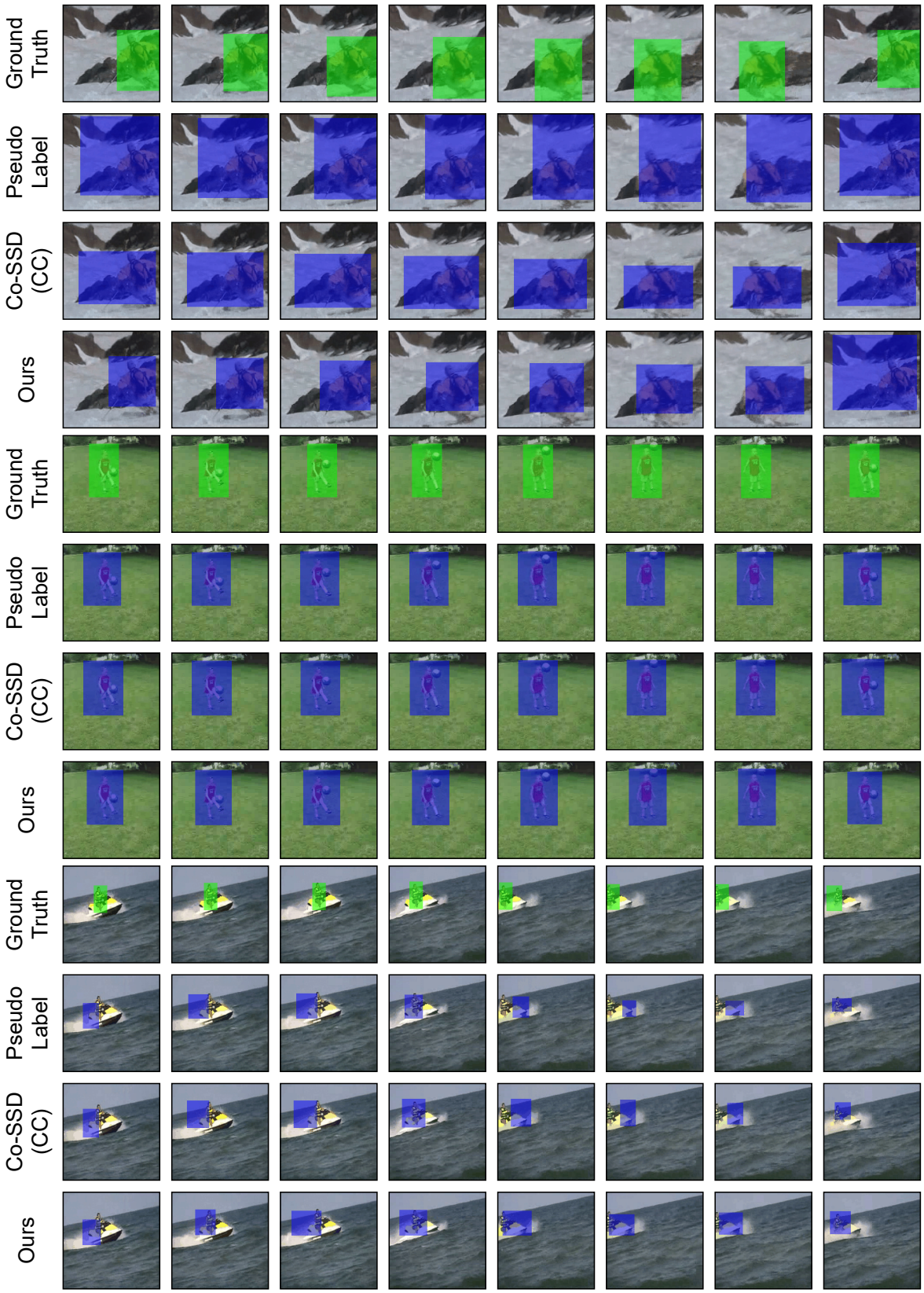


Figure 1. Performance for different semi-supervised approaches. The first column is the ground truth frame, second column depicts the ground truth localization, then further two columns show the prediction by the baseline semi-supervised approaches pseudo-label [3] and Co-SSD (CC) [4]. The final column is the overlap for our final approach.