Online Model Distillation for Efficient Video Inference (Supplementary)

1. Qualitative Comparison Videos

We include one minute video clips (videos.zip) from a selection of video streams in the LVS dataset, with Mask R-CNN and JITNet 0.9 predictions overlaid on the left and right respectively. All videos are subsampled by $4\times$ temporally to reduce file size. Full videos can be found at this anonymous YouTube channel: https://www.youtube.com/channel/UC-T0ferolHcDDKs2BZQEmrQ

2. Online Distillation Ablation Study

Our online distillation approach has several parameters (maximum updates (u_{max}), minimum stride (δ_{min}), learning rate and network size) that enable different trade-offs between accuacy and efficiency. Here, we study the impact of these parameters on the accuracy vs. efficiency trade-off on a subset of six video streams (which are representative of different scenarios) in the LVS dataset. We also evaluate the impact of skip connections and resolution on accuracy and efficiency. Table 1 compares the accuracy, speedup (relative to running the teacher on every frame), fraction of frames used for supervision, and number of FLOPS (floating point operations) for both training and inference on each of the variants. The baseline is JITNet 0.8, the online distillation algorithm run with an accuracy threshold of 0.8. For JITNet 0.8, the maximum updates, minimum stride, and learning rate were set to 8, 8 and 0.01 respectively. We vary one parameter at a time, and each column in the table corresponds to a variation of the JITNet 0.8 baseline.

Learning rate: High learning rates allow for faster adaptation. Therefore, we chose the highest learning rate at which online training is stable for all our experiments. As one can see in Table 1, a lower learning rate of 0.001 reduces both accuracy and speedup. Increasing the learning rate to 0.1 destabilizes training and yields low accuracy.

Max updates and stride: The number of updates needed on a single frame depends on how much the model can learn from one frame, and how useful that information is in the immediate future. Increasing the number of updates leads to overfitting, reducing accuracy while increasing speedup, and reducing the number of teacher samples used. This suggests some room for improvement in choosing how many updates to perform on a given frame over our simple accuracy-based heuristic. As one would expect, increasing and decreasing minimum stride increase and decrease accuracy respectively.

JITNet capacity: Intuitively, as the capacity of the student architecture is increased, the student model should require less help from the teacher. We verify this by varying the width of JITNet (the number of channels in each layer), and observe that a smaller capacity network (width 0.5) requires more supervision from the teacher, and also results in a significant drop in accuracy. Doubling JITNet width improves overall accuracy and reduces the number of teacher samples used. However, overall speedup is lower than the baseline due to the increased inference and training cost of the wider JITNet model.

JITNet resolution: High resolution is necessary for maintaining high accuracy on video streams that have small objects. When the input resolution to JITNet is halved (scale 0.5), there is an overall increase in the number of frames on which teacher supervision is used, and also a drop in accuracy relative to the baseline. However, reducing the resolution to 75% (scale 0.75) retains high accuracy while being slightly faster than the baseline. This suggests the possibility of varying resolution based on the contents in a video stream, which could be explored in the future.

Skip connections: We added encoder-to-decoder skip connections to facilitate better gradient propagation and make JITNet suitable for fast online adaptation. We evaluate the impact of the skip connections by removing them (Table 1, No Skip). JITNet without skip connections requires more teacher samples and adaptation, reducing both accuracy and speedup relative to the baseline.

Overall, the online distillation algorithm is reasonably robust to different parameter settings and provides a range of options for accuracy vs. efficiency.

3. MobileNet Student

We compare JITNet with a popular efficiency-oriented MobileNetV2 [5, 6] architecture in the context of online distillation. Table 1 shows online accuracy and speedup of online distillation when the MobileNetV2 architecture is used as the student. The two MobileNetV2 variants produce out-

	JITNet											MobileNet	
	Baseline	Max Updates		Learning Rate		Min Stride		Width		Skip	Scale		Output Stride	
		4	16	0.001	0.1	4	16	0.5	2.0	No	0.5	0.75	8	16
Accuracy (mIoU)	78.7	77.3	78.0	75.6	16.7	79.8	76.1	62.0	80.3	76.0	75.9	78.1	75.3	74.6
Speed Up	19.2×	17.2×	22.8×	14.3×	7.3×	17.1×	22.9×	14.9×	12.4×	12.4×	20.0×	22.0×	9.5×	16.8×
Teacher Samples	5.0%	6.1%	3.7%	6.7%	10.6%	7.7%	3.3%	5.1%	4.2%	6.3%	6.2%	5.3%	5.0%	5.8%
Inference (FLOPS)	15.2							11.8	47.9	14.3	4.6	10.3	60.3	18.3
Training (FLOPS)	42.0							31.8	140.4	39.4	8.6	22.4	176.1	53.0

Table 1: Comparison of different input parameter settings to the online distillation algorithm. The algorithm is robust to all parameter changes except very high learning rates, where online training becomes unstable.

Method	JM	JR	JD	FM	FR	FD
JITNet A	0.642	0.731	0.238	0.680	0.761	0.235
JITNet B	0.796	0.927	0.018	0.798	0.904	0.060
JITNet C	0.811	0.924	-0.004	0.831	0.913	0.004
OSVOS-S [3]	0.856	0.968	0.055	0.875	0.959	0.082
OSVOS [1]	0.798	0.936	0.149	0.806	0.926	0.150

Table 2: Accuracy comparison of different methods using the JITNet architecture and recent methods for semi-supervised video object segmentation on the DAVIS 2016 benchmark.

puts at $1/8^{th}$ and $1/16^{th}$ of the input resolution. As one can see, the higher resolution variant of the MobileNetV2 architecture is significantly slower and has lower accuracy than the JITNet baseline. Even the lower resolution (scale 0.75) version of JITNet has higher accuracy and speedup compared to the MobileNetV2 student. These results demonstrate that off-the-shelf models can be used in our online distillation framework. However, JITNet provides a better accuracy vs. efficiency spectrum compared to MobileNetV2 for online distillation, since it is designed for fast adaptation. Note that we measure FLOPS, which is a platform-agnostic metric, to ensure fair comparison, since wall-clock time (MobileNetV2 takes 38ms for inference compared to 7ms for JITNet on a Nvidia V100 GPU) depends on various factors, including target platform, underlying libraries, and specific implementation.

4. DAVIS Evaluation

Online distillation as a technique can be used to mimic an accurate teacher model with a compact model, improving runtime efficiency. The main focus of this work is to demonstrate the viability of the online distillation technique for semantic segmentation on streams captured from typical use case scenarios. In this section, we show preliminary results on the viability of online distillation combined with the JITNet architecture for accelerating semi-supervised video object segmentation methods. Specifically, we evaluate how the JITNet architecture can be combined with state-of-the-art methods such as OSVOS-S [3].

We evaluate three different configurations of JITNet at varying levels of supervision. In configuration A, we train JITNet on only the first ground truth frame of each sequence, and evaluate JITNet over the rest of the frames in the sequence without any additional supervision (the standard video object segmentation task). On many sequences in DAVIS, object appearance changes significantly and requires prior knowledge of the object shape. Note that JITNet is a very low capacity model designed for online training, and cannot encode such priors. Configuration A is not an online distillation scenario, but even with its low capacity, the JITNet architecture trained on just the first frame yields reasonable results.

Recent methods such as OSVOS-S [3] leverage instance segmentation models such as Mask R-CNN for providing priors on object shape every frame. We take a similar approach in configuration B, where the goal is to mimic the expensive OSVOS-S model. We train JITNet on the first ground truth frame, then adapt using segmentation predictions from OSVOS-S [3] every 16 frames. Note that in configuration B, our combined approach does not use additional ground truth, since OSVOS-S predictions are made using only the first ground truth frame. Finally, in configuration C, we train on the first ground truth frame, and adapt on the ground truth mask every 16 frames. This gives an idea of how the quality of the teacher effects online distillation.

We use the validation set of the DAVIS 2016 [4] dataset for our evaluation. The dataset contains 50 video sequences of 3455 frames total, each labeled with pixel-accurate segmentation masks for a single foreground object. We evaluate using the main DAVIS metrics: region similarity J and contour accuracy F, with precision, recall, and decay over time for both. We present metrics over the entire DAVIS 2016 validation set for all three JITNet configurations, alongside a subset of state-of-the-art video object segmentation approaches. In all configurations, we start with JITNet pre-trained on YouTube-VOS [8], with max updates per frame set to 500, accuracy threshold set to 0.95, and use

standard data augmentation (flipping, random noise, blurring, rotation). JITNet A performs similarly to OFL [7], a flow-based approach for video object segmentation, while JITNet B, using OSVOS-S predictions, performs comparably to OSVOS, with significantly lower runtime cost. Finally, JITNet C, which uses ground truth masks for adaptation, performs comparably to only using OSVOS-S predictions. This suggests that even slightly noisy supervision suffices for online distillation. Overall, these results are encouraging with regards to further work into exploring architectures well suited for online training.

5. Offline Training Details

JITNet COCO pre-training: All JITNet models used in our experiments are pre-trained on the COCO dataset. We convert the COCO instance mask labels into semantic segmentation labels by combining all the instance masks of each class for each image. We train the model on all 80 classes. The model is trained on 4 GPUs with batch size 24 (6 per GPU) using an Adam optimizer with a starting learning rate of 0.1 and a step decay schedule (reduces learning rate to 1/10th of current rate every 10 epochs) for 30 epochs.

JITNet offline oracle training: All offline oracle models are initialized using the COCO pre-trained model and trained on the specialized dataset for each video using the same training setup as COCO, i.e., same number of GPUs, batch size, optimizer, and learning rate schedule. However, each of the specialized datasets is about 6000 images, 20× smaller than the COCO dataset.

6. Standalone Semantic Segmentation

The JITNet architecture is specifically designed with low capacity so that it can support both fast training and inference. To understand the accuracy vs. efficiency trade-off relative to other architectures such as MobileNetV2 [5, 6], we trained a JITNet model with twice the number of channels and encoder/decoder blocks than the one used in the paper. This modified architecture is 1.5× faster than the semantic segmentation architecture based on MobileNetV2. The larger JITNet gives a mean IoU of 67.34 on the cityscapes [2] validation set and compares favorably with the 70.71 mean IoU of the MobileNetV2 based model [6]. We started with the larger JITNet architecture in the online distillation experiments, but lowered the capacity even further, with half the number of channels and encoder/decoder blocks, since it provided a better cost vs. accuracy trade-off for online distillation.

7. Additional Results

Table 3 gives the accuracy and performance of online distillation for each individual video stream we used in our evaluation, using JITNet at three different accuracy thresholds: JITNet 0.7, 0.8, and 0.9. Figure 1 shows the mean IoU of JITNet 0.8 and the offline oracle across time for additional videos streams. The top plot displays the mean IoU of both methods (data points are averages over 30 second time intervals). The bottom plot displays the number of JITNet model updates in each interval. Images above the plots are representative frames from time intervals requiring the most JITNet updates.

References

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 2

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 3

[3] K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. 2018. 2

[4] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3

[6] TensorFlow. TensorFlow DeepLab Model Zoo. https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md, 2018. 1, 3

[7] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[8] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2

Video	Offline	Flow		Online Distillation		
	Oracle (20%)	Slow (2.2×) (12.5%)	Fast (3.2×) (6.2%)	JITNet 0.7	JITNet 0.8	JITNet 0.9
Overall	80.3	76.6	65.2	75.5 (17.4×, 3.2%)	78.6 (13.5×, 4.7%)	82.5 (×7.5, 8.4%)
Category Averages						
Sports (Fixed)	87.5	81.2	71.0	80.8(36.7×, 1.6%)	82.8(33.3×, 1.8%)	87.6(16.1×, 5.1%)
Sports (Moving)	82.2	72.6	59.8	76.0(31.4×, 2.1%)	79.3(22.2×, 3.6%)	84.1(9.3×, 9.1%)
Sports (Ego)	72.3	69.4	55.1	65.0(21.1×, 3.7%)	70.2(14.1×, 6.0%)	75.0(7.7×, 10.4%)
Animals	89.0	83.2	73.4	82.9(33.1×, 1.9%)	84.3(30.1×, 2.2%)	87.6(22.0×, 4.4%)
Traffic	82.3	82.6	74.0	79.1(18.4×, 4.6%)	82.1(13.3×, 7.1%)	84.3(8.4×, 10.1%)
Driving/Walking	50.6	69.3	55.9	59.6(9.0×, 8.6%)	63.9(7.6×, 10.5%)	66.6(6.7×, 11.9%)
Individual Video Streams						
Badminton (P)	83.1	83.2	72.9	77.1(36.7×, 1.6%)	80.0(32.6×, 1.8%)	87.3(9.8×, 7.9%)
Squash (P)	88.4	70.0	56.5	80.9(37.0×, 1.6%)	82.5(35.7×, 1.7%)	86.0(21.3×, 3.2%)
Table Tennis (P)	89.4	84.8	75.4	81.5(37.2×, 1.6%)	83.5(36.7×, 1.6%)	88.3(20.3×, 3.4%)
Softball (P)	89.2	86.7	79.2	83.8(36.0×, 1.7%)	85.3(28.2×, 2.3%)	88.8(13.1×, 5.7%)
Hockey (P)	81.9	68.0	54.5	75.7(31.5×, 2.0%)	79.0(18.5×, 3.8%)	84.2(7.3×, 10.8%)
Soccer (P)	80.0	68.3	54.6	75.2(33.2×, 1.8%)	79.0(18.9×, 3.7%)	83.7(7.3×, 10.8%)
Tennis (P)	87.3	80.1	67.5	81.1(35.9×, 1.6%)	82.5(32.2×, 1.9%)	87.2(15.4×, 4.8%)
Volleyball (P)	82.3	82.9	73.0	76.4(34.3×, 1.7%)	80.3(21.1×, 3.2%)	85.0(8.4×, 9.2%)
Ice Hockey (P)	79.0	72.8	60.2	72.0(30.8×, 2.0%)	76.3(19.1×, 3.7%)	81.8(7.3×, 10.7%)
Kabaddi (P)	88.2	78.9	66.7	83.8(37.2×, 1.6%)	84.5(35.6×, 1.7%)	87.9(12.1×, 6.3%)
Figure Skating (P)	84.3	54.8	37.9	72.3(24.3×, 2.8%)	76.0(17.6×, 4.1%)	83.5(8.3×, 9.4%)
Drone (P)	74.5	70.5	58.5	70.8(23.7×, 2.8%)	76.6(10.7×, 7.2%)	79.9(6.3×, 12.5%)
Elephant (E)	93.3	91.0	85.3	92.7(37.1×, 1.6%)	92.8(37.2×, 1.6%)	93.6(36.6×, 1.6%)
Birds (B)	92.0	80.0	68.0	85.3(37.0×, 1.6%)	85.7(36.8×, 1.6%)	87.9(33.7×, 1.8%)
Giraffe (P,G)	85.5	79.6	69.2	82.8(32.1×, 1.9%)	84.1(26.4×, 2.5%)	87.6(11.4×, 6.6%)
Dog (P,D,C)	86.1	80.4	71.1	78.4(29.3×, 2.2%)	81.2(21.4×, 3.2%)	86.5(9.2×, 8.4%)
Horse (P,H)	87.9	84.9	73.4	75.3(30.1×, 2.1%)	77.7(28.6×, 2.2%)	82.7(19.2×, 3.6%)
Ego Ice Hockey (P)	68.8	56.7	39.6	56.3(31.1×, 2.0%)	59.3(20.1×, 3.4%)	67.0(7.8×, 10.0%)
Ego Basketball (P,C)	68.4	70.5	56.2	59.8(13.1×, 5.7%)	67.9(9.9×, 7.8%)	70.1(7.4×, 10.7%)
Ego Dodgeball (P)	82.1	75.5	60.4	74.3(26.6×, 2.5%)	79.5(20.3×, 3.4%)	84.2(9.5×, 8.2%)
Ego Soccer (P)	71.3	72.9	58.2	66.3(14.8×, 5.0%)	72.1(9.5×, 8.1%)	78.3(7.2×, 10.9%)
Biking (P,B)	70.7	71.6	61.3	68.2(19.8×, 3.5%)	72.3(10.4×, 7.3%)	75.3(6.4×, 12.4%)
Streetcam1 (P,C)	86.0	76.8	65.3	79.1(25.2×, 2.5%)	82.1(19.1×, 3.6%)	85.5(13.8×, 5.2%)
Streetcam2 (P,C)	82.2	82.1	72.9	76.1(15.9×, 4.6%)	79.7(10.1×, 7.6%)	83.7(6.5×, 12.2%)
Jackson Hole (P,C)	76.5	77.9	67.9	75.7(12.8×, 5.9%)	78.0(9.4×, 8.3%)	79.2(7.4×, 10.7%)
Murphys (P,C,B)	91.9	94.1	91.2	88.0(32.1×, 1.9%)	89.8(26.0×, 2.5%)	92.9(9.9×, 7.8%)
Samui Street (P,C,B)	80.6	83.8	76.5	78.8(13.6×, 5.5%)	82.6(8.2×, 9.5%)	83.7(6.5×, 12.2%)
Toomer (P,C)	76.6	81.1	70.4	76.9(10.7×, 7.2%)	80.3(7.0×, 11.3%)	80.5(6.4×, 12.4%)
Driving (P,C,B)	51.1	72.2	59.7	63.8(8.9×, 8.8%)	68.2(6.9×, 11.5%)	66.7(6.4×, 12.4%)
Walking (P,C,B)	50.2	66.4	52.1	55.4(9.1×, 8.5%)	59.6(8.2×, 9.5%)	66.4(7.0×, 11.3%)

Table 3: Comparison of accuracy (mean IoU over all the classes excluding background), runtime speedup relative to MRCNN (where applicable), and the fraction of frames where MRCNN is run. Classes present in each video are denoted by letters (A - Auto, Bi - Bird, Bk - Bike, D - Dog, E - Elephant, G - Giraffe, H - Horse, P - Person). Overall, online distillation using JITNet provides a better accuracy/efficiency tradeoff than baseline methods.

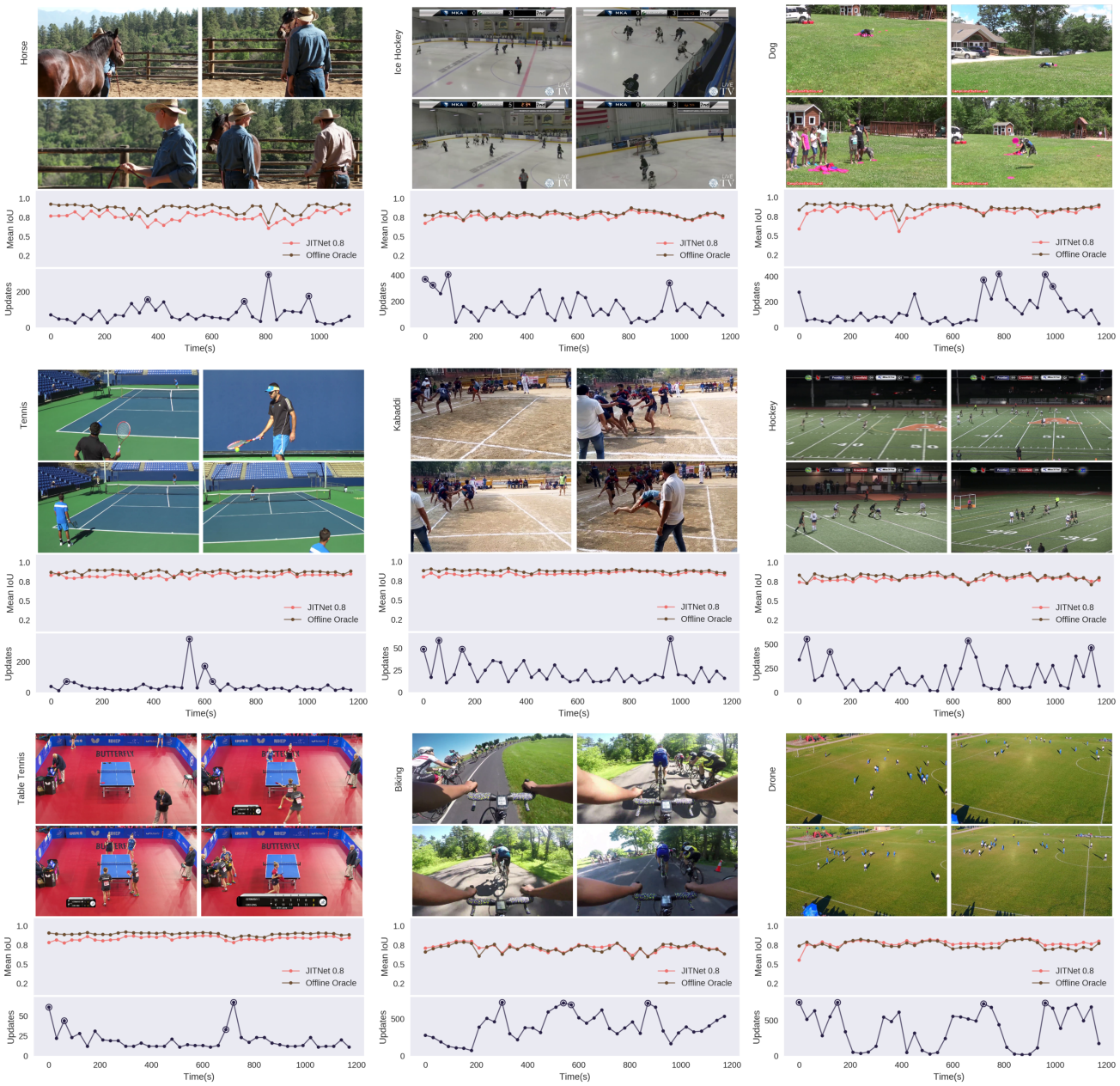

Figure 1: Top graph: the accuracy of JITNet 0.8 and Offline Oracle relative to MRCNN. Bottom graph: the number of updates to JITNet during online distillation. Plotted points are averages over a 30 second interval of the video. Images correspond to circled points in bottom plot, and show times where JITNet required frequent training to maintain accuracy.