OpenTAL: Towards Open Set Temporal Action Localization Supplementary Material

Wentao Bao, Qi Yu, Yu Kong
Rochester Institute of Technology, Rochester, NY 14623, USA
{wb6219, qi.yu, yu.kong}@rit.edu

In this document, we provide the detailed proof of the gradient of the EDL loss (Sec. A), the dataset description of the open set setting (Sec. B), implementation details (Sec. C), additional results and discussions (Sec. D).

A. Gradient of EDL

Given the DNN logits $\mathbf{z}_i \in \mathbb{R}^K$ of sample x_i, an evidence function defined by exp is applied to the logits to get the class-wise evidence prediction, i.e., $\mathbf{e}_i = \exp(\mathbf{z}_i)$. Following the maximum likelihood loss form of Evidential Deep Learning (EDL) [9], we have the EDL loss:

$$\mathcal{L}^{(i)}_{\text{EDL}}(\boldsymbol{\alpha}_i) = \sum_{j=1}^{K} t_{ij}(\log(S_i) - \log(\alpha_{ij})), \tag{1}$$

where $t_{ij} = 1$ iff. the class label $y_i = j$, otherwise $t_{ij} = 0$. The total Dirichlet strength $S_i = \sum_j \alpha_{ij}$ and the class-wise strength $\boldsymbol{\alpha}_i = \mathbf{e}_i + 1$. Therefore, according to the simple chain rule, we have the partial derivative:

$$\frac{\partial \alpha_{ij}}{\partial z_{ij}} = \frac{\partial \alpha_{ij}}{\partial e_{ij}} \cdot \frac{\partial e_{ij}}{\partial z_{ij}} = e_{ij} \tag{2}$$

Then, the gradient of the j-th entry in Eq. (1), i.e., $\mathcal{L}^{(ij)}_{\text{EDL}}$, w.r.t. the logits z_{ij} can be derived as follows:

$$\begin{aligned} g_{ij} = \frac{\partial \mathcal{L}^{(ij)}_{\text{EDL}}}{\partial z_{ij}} &= t_{ij}\left[\frac{1}{S_i}\frac{\partial S_i}{\partial z_{ij}} - \frac{1}{\alpha_{ij}}\frac{\partial \alpha_{ij}}{\partial z_{ij}}\right] \\ &= t_{ij}\left[\frac{1}{S_i}\sum_{k=1}^{K}\frac{\partial \alpha_{ik}}{\partial z_{ij}} - \frac{e_{ij}}{\alpha_{ij}}\right] \\ &= t_{ij}\left[\frac{1}{S_i}\sum_{k=1}^{K} e_{ik} - \frac{e_{ij}}{\alpha_{ij}}\right] \end{aligned} \tag{3}$$

Consider that $S_i = \sum_k \alpha_{ik} = \sum_j e_{ik} + K$, and the evidential uncertainty $u_i = K/S_i$, we further simplify the g_{ij} as follows:

$$\begin{aligned} g_{ij} &= t_{ij}\left[\frac{S_i - K}{S_i} - \frac{\alpha_{ij} - 1}{\alpha_{ij}}\right] \\ &= t_{ij}\left[\frac{S_i - K\alpha_{ij}}{S_i\alpha_{ij}}\right] \\ &= t_{ij}\left[\frac{1}{\alpha_{ij}} - u_i\right], \end{aligned} \tag{4}$$

which has proved the equation of g_{ij} in our main paper. From this conclusion, when considering that $\alpha_{ij} \in (1, \infty)$ and $u_i \in (0, 1)$, we have the property $|g_{ij}| \in [0, 1)$.

Furthermore, consider the last DNN layer parameters $\mathbf{w} \in \mathbb{R}^{D\times K}$ such that $\mathbf{z}_i = \mathbf{w}^T\mathbf{h}_i$ where $\mathbf{h}_i \in \mathbb{R}^D$ is the high-dimensional feature of x_i, we can derive the gradient of EDL loss w.r.t. parameters $\mathbf{w}$:

$$\nabla_w \mathcal{L} = \frac{\partial \mathcal{L}^{(ij)}_{\text{EDL}}}{\partial w_{dk}} = \frac{\partial \mathcal{L}^{(ik)}_{\text{EDL}}}{\partial z_{ik}} \cdot \frac{\partial z_{ik}}{\partial w_{dk}} = g_{ik} \cdot h_{id}, \tag{5}$$

where w_{dk} and h_{id} are elements of the matrix $\mathbf{w}$ and the vector $\mathbf{h}_i$. Similar to [8], we consider the influence function [6] by ignoring the inverse of Hessian and using the magnitude (L_1 norm) of the gradient:

$$\begin{aligned} \omega_i = \|\nabla_w \mathcal{L}\|_1 &= \sum_{k=1}^{K}\sum_{d=1}^{D}|g_{ik} \cdot h_{id}| \\ &= \left(\sum_{k=1}^{K}|g_{ik}|\right)\left(\sum_{d=1}^{D}|h_{id}|\right) \\ &= \|\mathbf{g}_i\|_1 \cdot \|\mathbf{h}_i\|_1, \end{aligned} \tag{6}$$

which has proved the equation of ω_i in our main paper.

B. Dataset Details

To enable the existing Temporal Action Localization (TAL) datasets such as THUMOS14 [5] and ActivityNet1.3 [3] for the open set TAL setting, a subset of action categories has to be reserved as the unknown used in

Table 1. **THUMOS14 Splits for Open Set TAL.** For each split, five out of twenty action categories are randomly selected as the unknown (**U**) used in open set testing, while the rest fifteen categories are the known (**K**) used in model training.

	Split 1	Split 2	Split 3
BaseballPitch	K	K	K
BasketballDunk	K	K	K
Billiards	K	K	K
CricketBowling	K	U	K
CricketShot	K	K	U
FrisbeeCatch	K	K	K
GolfSwing	K	K	K
HammerThrow	K	U	K
HighJump	K	K	K
JavelinThrow	K	U	U
PoleVault	K	K	U
Shotput	K	K	U
TennisSwing	K	K	K
ThrowDiscus	K	K	K
VolleyballSpiking	K	K	K
CleanAndJerk	U	K	K
CliffDiving	U	U	K
Diving	U	U	K
LongJump	U	K	U
SoccerPenalty	U	K	K

open set testing. In practice, we randomly splitted the THUMOS14 three times into known and unknown subsets of categories. For each split, a model will be trained on the closed set (which only contains known categories), and tested on the open set that contains both known and unknown categories. Table 1 shows the detailed information of the three dataset splits from THUMOS14.

To further increase the openness in testing, we incorporate activity categories from ActivityNet1.3 that are non-overlapped with THUMOS14 into the open set testing. Specifically, the following 14 overlapping activity categories are removed: *Table soccer, Javelin throw, Clean and jerk, Springboard diving, Pole vault, Cricket, High jump, Shot put, Long jump, Hammer throw, Snatch, Volleyball, Plataform diving, Discus throw*. Note that we did not use ActivityNet1.3 for similar model training as the THUMOS14, e.g., train a model on multiple random splits of ActivityNet1.3, due to the limited computational resource.

C. Implementation Details

Detailed Architecture The proposed OpenTAL is primarily implemented on the AFSD [7] framework. It uses a pre-trained I3D [4] as the feature extraction backbone and a 6-layer temporal FPN architecture is applied to the I3D for action classification and localization. Each level consists of a coarse stage, a saliency-based proposal refinement module, and a refined stage. The first two pyramid levels use 3D convolutional (Conv3D) block while the rest four levels use 1D convolutional (Conv1D) block. Group Normalization and ReLU activation are utilized in each block. The temporal localization head and action classification head are implemented by a shared Conv1D block across all 6 levels. To implement OpenTAL method, the $(K+1)$-way classification head is replaced with K-way evidential neural network head, while the localization head is kept unchanged. We additionally add an actionness prediction branch which consists of a Conv1D block for both the coarse and the refined stages.

Training and Testing In training, the proposed classification loss $\mathcal{L}_{\text{MIB-EDL}}$ and actionness prediction loss $\mathcal{L}_{\text{ACT}}$ are applied to both the coarse and refined stages in AFSD, while the IoU-aware uncertainty calibration loss $\mathcal{L}_{\text{IoUC}}$ is only applied to the refined stage because this loss function is dependent of the pre-computed temporal IoU using the predicted action locations in the coarse stage. Similar to AFSD, we used temporal IoU threshold 0.5 in the training to identify the foreground actions from the proposals. Besides, we reduced the weight of triplet loss in AFSD to 0.001 since the contrastive learning loss would not work well when there are unknown action clips in the background. The whole model is trained by Adam optimizer with base learning rate 1e-5 and weight decay 1e-3. All models are trained with 25 epochs to ensure full convergence and the model snapshot of the last epoch is used for testing and evaluation.

In testing, the actionness score is multiplied to the confidence score before the soft-NMS post-processing module. The σ and top-N hyperparameters are set to 0.5 and 5000, which are recommended by the AFSD.

D. Additional Results

Impact of tIoU Thresholds Since the proposed OSTAL task cares not only the classification but also the temporal localization, we present the experimental results under different temporal IoU (tIoU) thresholds. Following existing TAL literature, we set five tIoU thresholds $[0.3:0.1:0.7]$ when the unknown classes are from THUMOS14 and ten tIoU thresholds $[0.5:0.05:0.95]$ when the unknown classes are from ActivityNet1.3, respectively. Evaluation results by AUROC, AUPR, and OSDR are reported in Table 2, 3, and 4, respectively. The results show that AUROC performances are stable across different tIoU thresholds, while the AUPR and OSDR performances vary significantly as the tIoU threshold changes. Besides, as the tIoU threshold increasing, AUROC and OSDR values would increase accordingly. For all those tIoU thresholds and evaluation metrics, the proposed OpenTAL could consistently outperform baselines.

Table 2. **AUROC Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	54.70	55.46	56.41	57.12	57.11	56.16	56.97	58.41	55.97	57.77
OpenMax [2]	53.26	52.1	52.13	51.89	52.53	52.38	51.24	52.39	49.13	51.59
EDL [1]	64.05	64.27	65.13	66.21	66.81	65.29	62.82	66.23	67.92	65.69
OpenTAL	**78.33**	**79.04**	**79.30**	**79.40**	**79.82**	**79.18**	**82.97**	**83.21**	**83.38**	**83.22**

Table 3. **AUPR Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	31.85	31.81	31.11	29.78	27.99	30.51	53.54	44.15	34.54	44.77
OpenMax [2]	33.17	31.61	30.59	29.15	28.45	30.60	54.88	48.37	40.07	48.48
EDL [1]	40.05	39.45	38.05	37.58	36.35	38.30	53.97	47.22	45.59	48.46
OpenTAL	**58.62**	**59.40**	**58.78**	**57.54**	**55.88**	**58.04**	**80.41**	**74.20**	**73.92**	**75.54**

Table 4. **OSDR Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	23.40	25.19	27.43	29.97	32.08	27.61	27.63	33.73	31.59	32.01
OpenMax [2]	13.66	14.58	15.91	17.71	20.41	16.45	15.73	21.49	18.07	19.35
EDL [1]	36.26	37.58	39.16	41.18	42.99	39.43	38.56	43.72	42.20	42.18
OpenTAL	**42.91**	**46.19**	**49.50**	**52.50**	**56.78**	**49.57**	**50.49**	**59.87**	**62.17**	**57.89**

Impact of Dataset Splits For open set problems, splitting an existing fully annotated dataset into known and unknown part plays an important role in performance evaluation. In this document, we comprehensively show the ROC, PR, and OSDR curves on three different THUMOS14 open set splits in Fig. 1, Fig 2, and Fig. 3, respectively. From these figures, we can find that the ROC curves vary much more across different splits than tIoU thresholds, while the PR and OSDR curves vary significantly both across the splits and tIoU thresholds. Besides, for all sub-figures, the proposed OpenTAL could significantly outperform baselines.

More Visualizations In this document, we add more visualizations for comparing the proposed OpenTAL with baselines in Fig 4. The first 4 examples (in the first 2 rows) show that OpenTAL could well localize and recognize known actions (colorful segments). The rest of 12 examples show that OpenTAL can roughly localize and reject the unknown actions (black segments).

References

[1] Wentao Bao, Qi Yu, and Yu Kong. Evidential deep learning for open set action recognition. In *ICCV*, 2021. 3

[2] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *CVPR*, 2016. 3

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1

[4] J. Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017. 2

[5] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 1

[6] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017. 1

[7] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, 2021. 2

[8] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *ICCV*, 2021. 1

[9] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018. 1

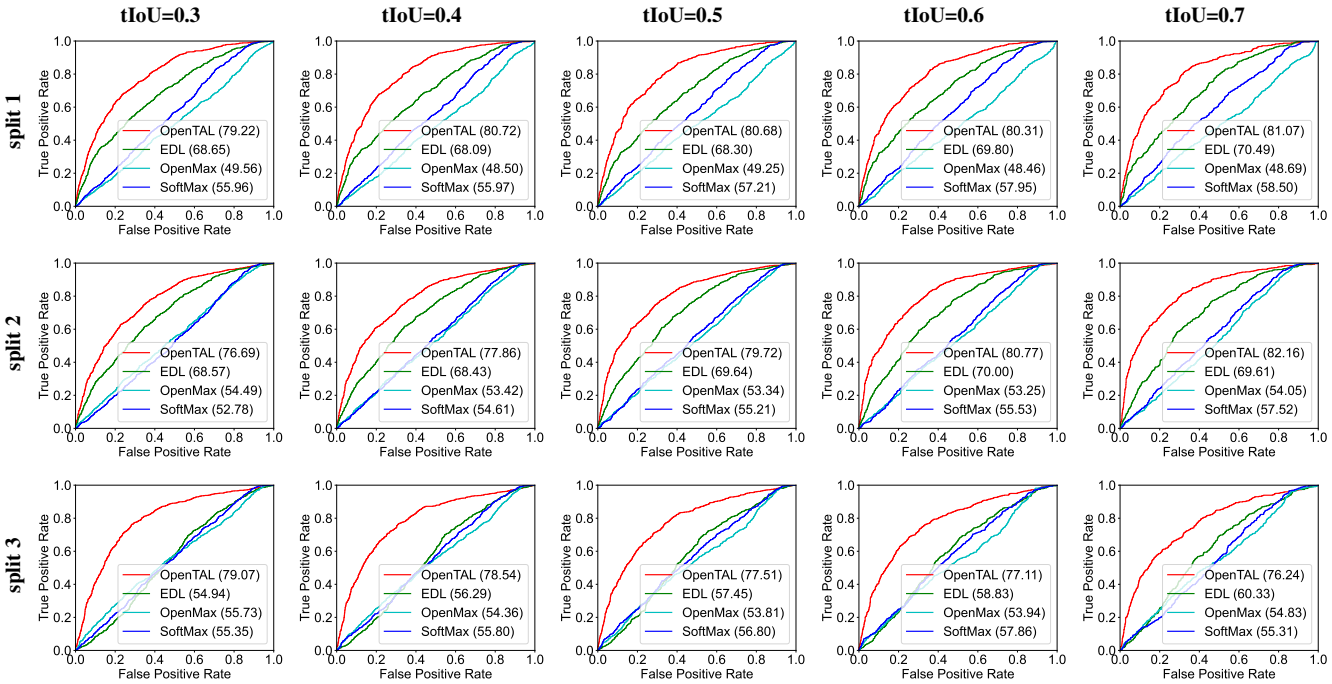

Figure 1. **ROC Curves.** These figures show the method comparison by ROC curves on THUMOS14 open set splits. Numbers in parentheses are AUROC values. They show that the ROC performance varies more across dataset splits than tIoU thresholds, and our proposed OpenTAL could consistently outperform baselines on all the three splits and five thresholds.

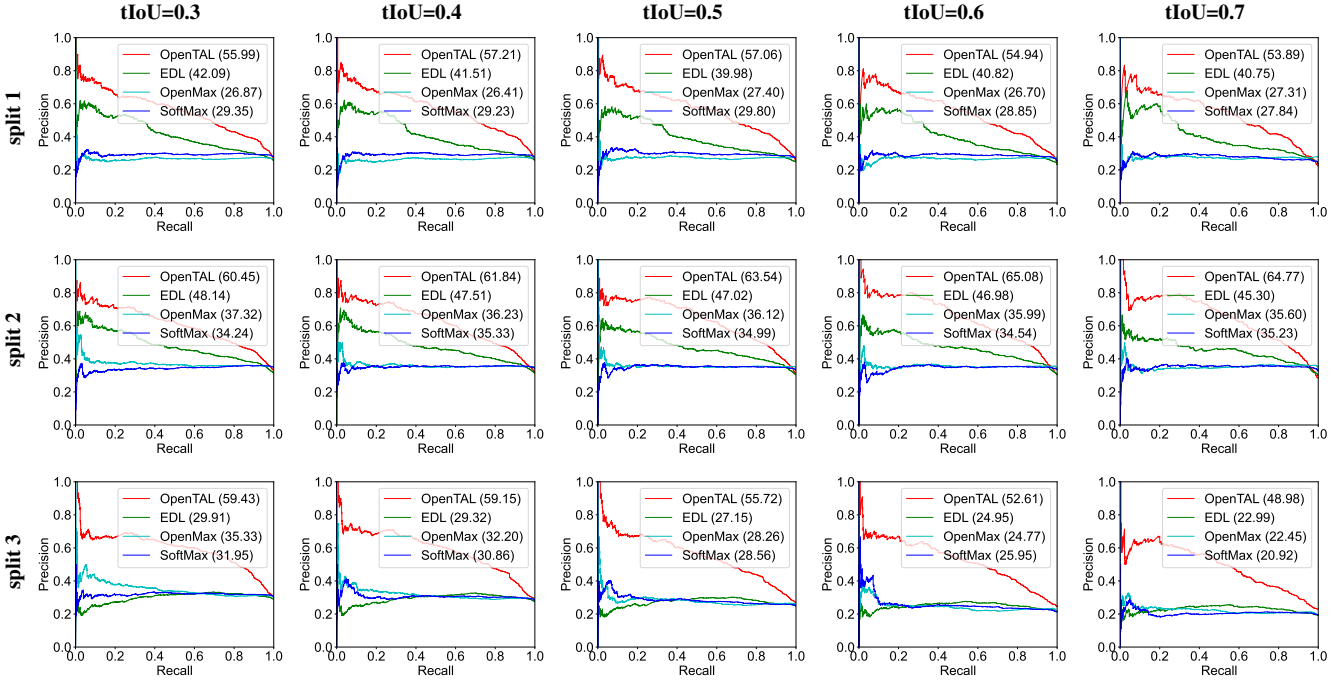

Figure 2. **PR Curves.** These figures show the method comparison by Precision Recall curves on THUMOS14 open set splits. Numbers in parentheses are AUPR values. They show that the PR performance varies significantly both across dataset splits and tIoU thresholds, and our proposed OpenTAL could consistently outperform baselines on all the three splits and five thresholds.

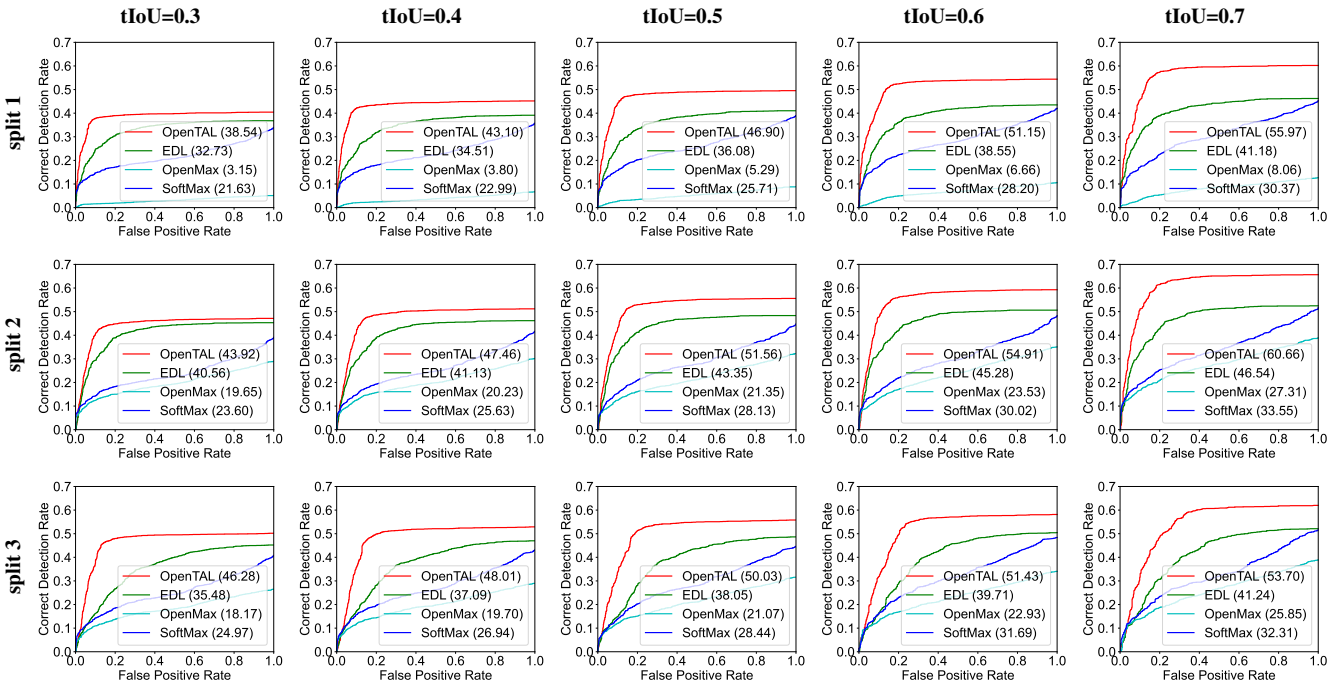

Figure 3. **OSDR Curves.** These figures show the method comparison by OSDR curves on THUMOS14 open set splits. Numbers in parentheses are OSDR values. They show that the OSDR performance varies significantly both across dataset splits and tIoU thresholds, and our proposed OpenTAL could consistently outperform baselines on all the three splits and five thresholds.

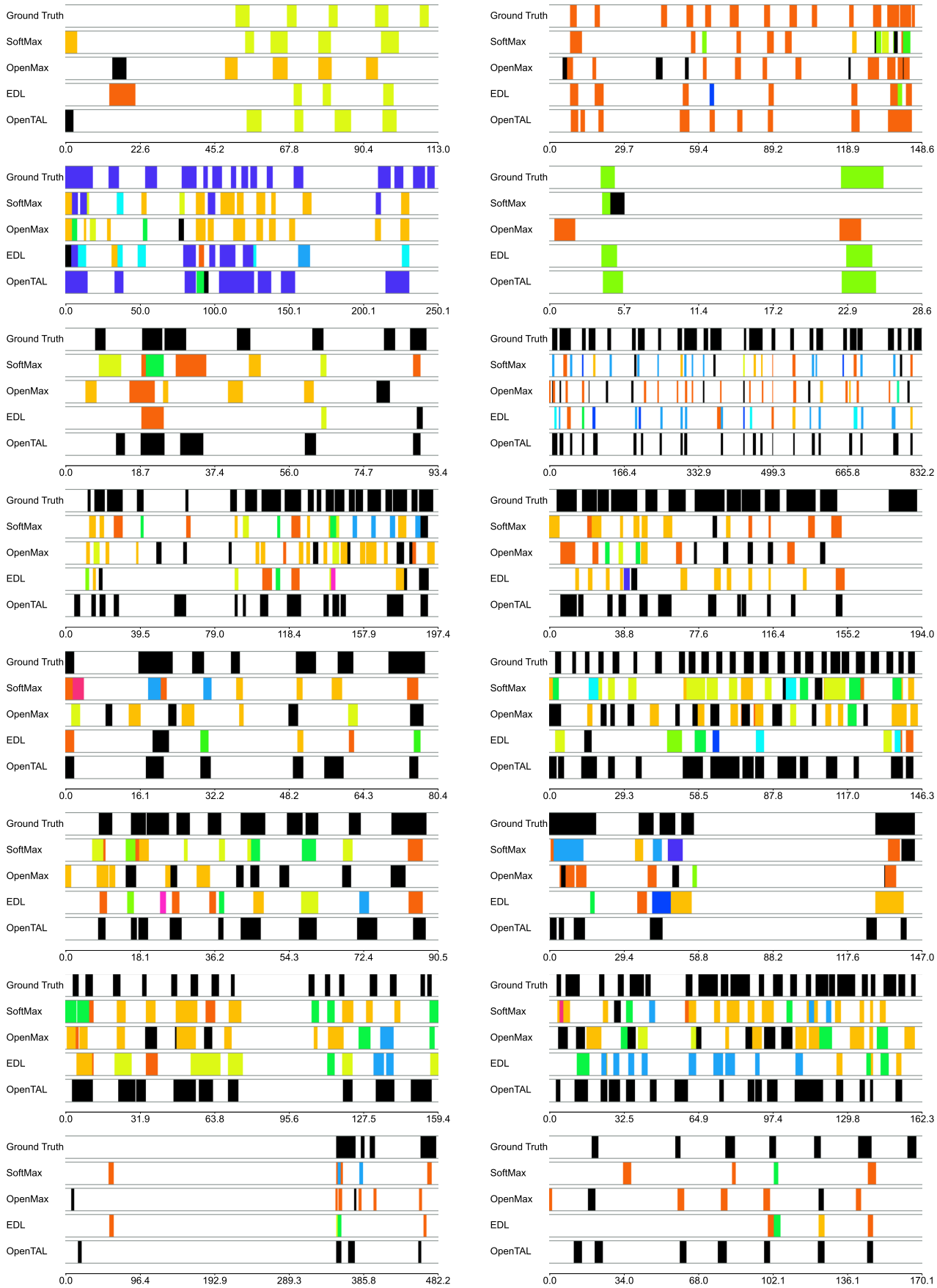

Figure 4. **Qualitative Results.** We show the actions of unknown classes with black color, while the rest colors are actions of known classes. The x-axis represents the timestamps (seconds).