Appendix for *Tragedy Plus Time*: Capturing Unintended Human Activities from Weakly-labeled Videos

Arnav Chakravarthy, Zhiyuan Fang, Yezhou Yang
Arizona State University
achakr37@asu.edu, zy.fang@asu.edu, yz.yang@asu.edu

1. Overview

This document provides additional details and further analysis of our model architecture. We start by providing detailed statistics about the W-Oops dataset in Sec. 2. We further analyse the dependence of unintentional actions on goal-directed actions in Sec 3. We then give more details on the choice of our backbone features and experimenting with human-pose features in Sec. 4 . We further analyse the video embedding module by removing it entirely/replacing it with a Transformer Encoder [16] in Sec. 5. We study the effect of different selections of λ, the hyperparameter that controls the trade-off between our losses in Sec. 6. Details into the 3D-CNN feature extraction is provided in Sec. 7. Finally, we explain more about our annotation tool in Sec. 8, and provide additional qualitative results for the localization and captioning experiments in Sec. 9 and Sec. 10.

2. W-Oops Statistics and Analysis

The final W-oops dataset contains 1582 train samples and 526 testing samples, containing a total of 44 diverse goal-directed and 30 unintentional action classes, as seen in Fig. 1. We have also provided the distribution of the goal-directed and unintentional segment lengths, as well as the total video lengths. It shows that the goal-directed and unintentional segment lengths are well diversified over then entire length of the video. The lengths of the video are short in general, with a majority of them ranging from 6.2 - 7.7 seconds. This makes the task of identifying these sub-regions in the video challenging. In our benchmark, train samples contain only video-level labels whereas the test samples contain both the video-level labels as well as the unintended activity transition points (taken from the original Oops dataset), which we use to split the video into a goal-directed and unintentional region in order, for evaluation.

3. Can Unintentional Actions be predicted knowing the Goal-Directed Action?

In this section we analyse the amount of information knowing about a goal-directed action gives us when inferring the unintentional action. In order to do this, we calculate a probability distribution of the unintentional actions conditioned on the goal-directed actions and calculate their entropy. An entropy of 0 would indicate that the unintentional action can predicted from the goal-directed action alone. On the other hand, an entropy of $4.91(-\log_2(30))$ indicates that the unintentional actions are uncorrelated with the goal-directed action. Fig. 2 shows us that the conditional entropy of unintentional actions lies between these two values, suggesting that they are correlated but are not completely predictable knowing the goal-directed action.

4. Using 2D Pose Features

Successful attempts at using human skeleton features for activity recognition. [10, 17, 20], fall prediction [7, 14] and action localization [11] provides encouragement to use them for our task as well. However human skeleton features alone would not be enough as it does not capture the surrounding environment information which the RGB features do. Hence we concatenate both the RGB features and skeleton features to use as our backbone features.

In order to test this hypothesis, for each video we extract 2D keypoint coordinates of human(s) from each observed frame using OpenPose [1]. Since OpenPose is able to capture multiple human(s) in a frame, we use DeepSort [18] to cluster the keypoints of the same person across frames. We denote the sequence of observed keypoints from the i^{th} person in the video as $\mathbf{K^i} = (\mathbf{k}_1^i, \mathbf{k}_2^i, ..., \mathbf{k}_t^i)$, where $\mathbf{k}_j^i$ denotes the keypoint coordinates of the i^{th} person in frame j, with t being the total number of frames. Example showed in 3

Using the COCO model of OpenPose, we obtain 18 keypoint coordinates for each observed person in a frame, which include coordinates for the nose, neck, left and right shoulders, hips, elbows, wrists, knees, ankles, eyes and ears,

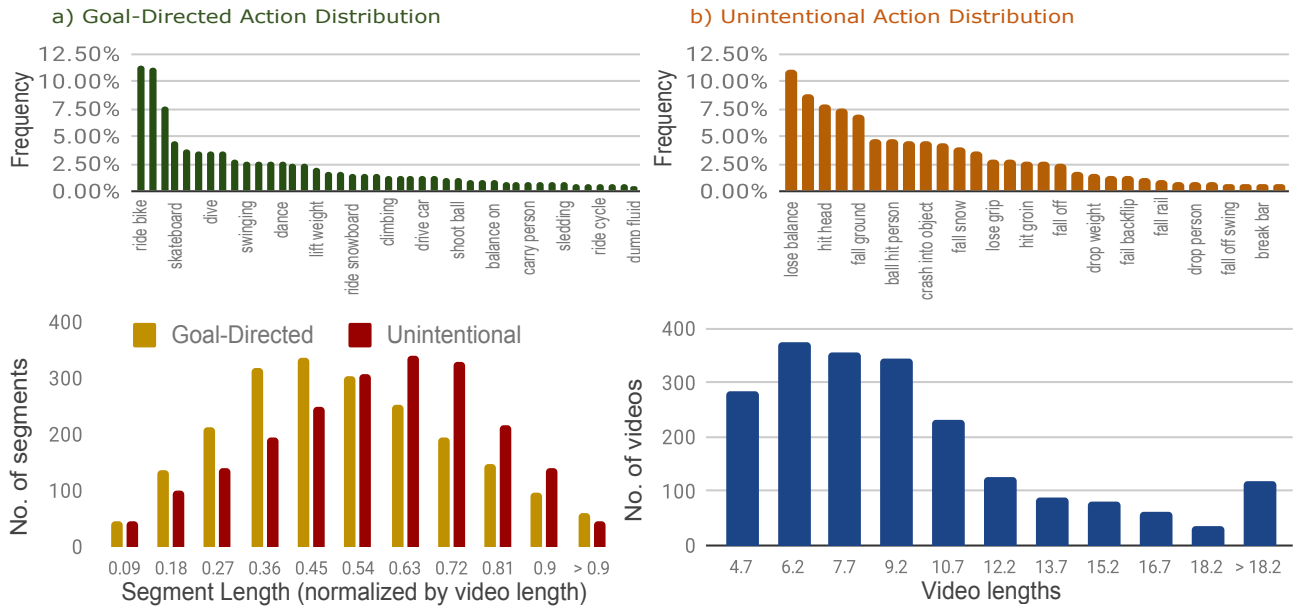

Figure 1. Top: Distribution over the goal-directed and unintentional actions (partially shown). Lower Left: Distribution over goal-directed and unintentional segment lengths (normalized by the video length). Lower Right: Distribution over the entire video length.

i.e., each

$$\mathbf{k}_j^i = (x_{j,1}^i, y_{j,1}^i, x_{j,2}^i, y_{j,2}^i, ..., x_{j,18}^i, y_{j,18}^i) \quad (1)$$

Since these coordinates do not capture the correlation between different keypoints, we follow the process in [7] to vectorize these coordinates to incorporate these correlations. We ignore the face keypoints (eyes, ears and nose), since we want to focus only on the body pose. We then transform the remaining 13 coordinates into vectors connecting the adjacent keypoints as illustrated in Fig. The shoulders are connected to the neck, elbows are connected to the corresponding shoulders, wrists are connected to corresponding elbows, hips to the neck, knees to the corresponding hips and finally the ankles to the corresponding knees. Following this process as followed in [7], we obtain 12 keypoint vectors from the 13 keypoint coordinates, and normalize them to unit length. For the m^{th} connection pointing from the p^{th} keypoint to the q^{th} keypoint, the keypoint vector $(\overline{x_{j,m}^i}, \overline{y_{j,m}^i})$ for the i^{th} person in frame j is calculated as:

$$(\overline{x_{j,m}^i}, \overline{y_{j,m}^i}) = \frac{(x_{j,q}^i - x_{j,p}^i, y_{j,q}^i - y_{j,p}^i)}{\sqrt{(x_{j,q}^i - x_{j,p}^i)^2 + (y_{j,q}^i - y_{j,p}^i)^2}} \quad (2)$$

We calculate this for each of the 12 connections, and concatenate them to get:

$$\overline{\mathbf{k}_j^i} = (\overline{x_{j,1}^i}, \overline{y_{j,1}^i}, \overline{x_{j,2}^i}, \overline{y_{j,2}^i}, ..., \overline{x_{j,12}^i}, \overline{y_{j,12}^i}) \quad (3)$$

Videos involving action such as two people colliding with another person, or a person carrying another person, requires features of multiple people in order to understand these actions. Hence we concatenate the keypoints of the two most frequently occurring people l and r as detected by DeepSort, and concatenate them to get the final feature vector for frame j as $\overline{\mathbf{k}_j} = \overline{\mathbf{k}_j^l} \oplus \overline{\mathbf{k}_j^r}$.

Note, that there may be partially missing or completely missing keypoint coordinates for a person in a certain frame. In the case of partially missing keypoints we set a keypoint vector containing a connection to a missing keypoint to (0,0). In the case of completely missing keypoints we set all the keypoint vectors to (0,0) in the case the person had not been detected yet, or else set all the keypoint vectors to the corresponding last observed keypoint vectors of the person.

RGB features are extracted by passing non-overlapping chunks of 16 frames to a pretrained 3D CNN architecture. Since the skeleton feature are extracted for each frame, we concatenate skeleton features extracted from consecutive and non-overlapping chunks of 16 frames. We convert $\overline{\mathbf{k}} = (\overline{\mathbf{k}_1}, \overline{\mathbf{k}_2}, ..., \overline{\mathbf{k}_t})$ to $\widetilde{\mathbf{k}} = (\widetilde{\mathbf{k}_1}, \widetilde{\mathbf{k}_2}, ..., \widetilde{\mathbf{k}_{t/16}})$, where $\widetilde{\mathbf{k}_h}$ for the h^{th} chunk is given by :

$$\widetilde{\mathbf{k}_h} = \overline{\mathbf{k}_{16(h-1)+1}} \oplus \overline{\mathbf{k}_{16(h-1)+2}} \oplus ... \oplus \overline{\mathbf{k}_{16(h)}} \quad (4)$$

We finally concatenate the RGB features X and the skeleton features $\widetilde{\mathbf{k}}$ to obtain $X_{cat} = (X_1 \oplus \widetilde{\mathbf{k}_1}, X_2 \oplus \widetilde{\mathbf{k}_2}), ..., X_l \oplus \widetilde{\mathbf{k}_l})$, where l is the total number of 16 frame chunks (clips) in the video.

We then provide comparisons between using only the RGB features and using the RGB features concatenated

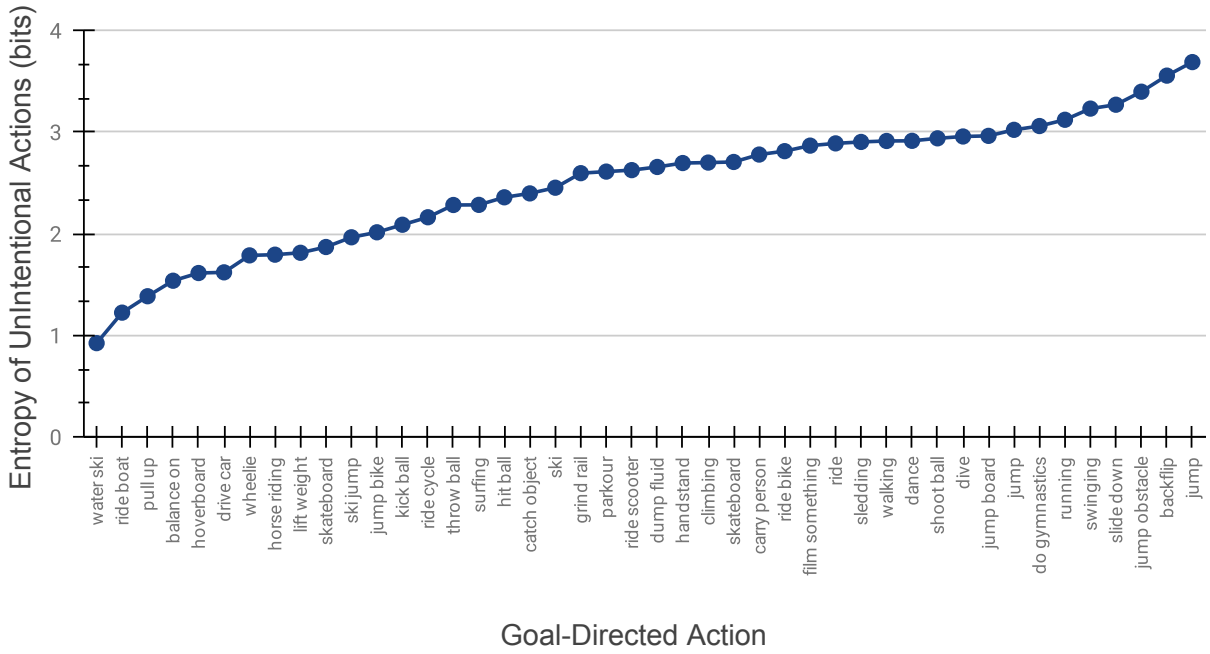

Figure 2. Entropy (in bits) of the unintentional actions conditioned on the goal-directed actions. We can see that the unintentional actions are correlated to the goal-directed actions but are not completely predictable.

with the skeleton features in table 1 . We can see that the performance decreases, from 35.0% to 34.7% for the goal-directed mAP@IoU and from 26.0% to 24.6% for the unintentional mAP@IoU. We conjecture that this performance decrease is due to the noise introduced by the incorrect/missing keypoint coordinates at certain frames, as well as due to some of the videos which involve an agent driving a vehicle and hence the agent is partially or completely not seen in the video

Feature	Segment	mAP@IoU			
		0.3	0.5	0.9	Avg
RGB (I3D)	Goal	49.9	41.1	5.0	35.0
	UnInt	36.4	30.0	2.8	26.0
RGB (I3D) + Skeleton	Goal	47.1	42.1	4.8	34.7
	UnInt	34.4	27.4	2.1	24.6

Table 1. Analysis of the effect of skeleton features.

5. Analysis of Video Embedding Module

We now analyse the effectiveness of our video embedding module, by removing the module and using only the raw features from the frozen feature extractor. We also compare our video embedding module which consists of a GRU with a Transformer Encoder [16], a component of the original Transformer architecture which has achieved state of the art results on many vision [2, 5, 22–25] as well as NLP [4, 9, 19, 21] tasks. As opposed to a GRU which learns feature representations at each time step in a sequential manner by using the hidden state in the previous timestep, a transformer encoder uses multiheaded self attention to calculate the dependency of each token in the sequence to encode the token at the current timestep. As seen in table 2, we can see that using static backbone features result in a very poor localization performance. Additionally it is also interesting to observe that the GRU performs better than the transformer.

Embedding Module	Segment	mAP@IoU			
		0.3	0.5	0.9	Avg(0.1:0.9)
None	Goal	30.2	16.5	1.3	18.7
	UnInt	18.6	9.4	0.02	11.1
Transformer Encoder	Goal	49.1	41.5	2.7	34.9
	UnInt	31.7	17.9	0.7	22.7
GRU	Goal	49.9	41.1	5.0	35.0
	UnInt	36.4	30.0	2.8	26.0

Table 2. Ablation study of the contribution of the video embedding module.

6. Analysis of Weight Tradeoff Parameter λ

λ is the scalar parameter used to control the tradeoff between the Multiple Instance Learning Loss (MIL) and the

Figure 3. An example of extracting body keypoint coordinates of multiple agents in videos using Openpose [1], followed by Deepsort [18] to cluster the keypoints of the same person across the frames.

Overlap Regularization. We study the effects of changing this parameter in the range of [0,1], where λ=0 corresponds to purely MIL Loss and λ=1 corresponds to purely Overlap Regularization. As seen in Fig. 4, we notice that for $0.3 \leq \lambda \leq 0.8$, the average mAP@IoU for the goal-directed and unintentional action remains almost constant, but on close observation we see that $\lambda = 0.8$ performs the best for the goal-directed as well as unintentional action.

7. Feature Extraction Details

This section provides detailed explanation about the feature extraction process. We follow previous work [6] and down-sample all raw videos at 25 FPS. We then create chunks of 16 consecutive and non-overlapping frames. In order the extract the I3D and R(2+1)D features, we pass these chunks to the respective backbone networks and obtain the features as the output of their global pooling layers. We use the following libraries to extract R(2+1)D[1] and I3D[2] features from the videos.

I3D: For the I3D [3] features, we re-scale all frame pixels between -1 and 1, after which we resize the frames preserving aspect ratio such that the smallest dimension is 256 pixels. We then apply center crop to obtain 224×224 frames. Chunks of 16 non-overlapping frames are then passed through the RGB stream of a I3D [3] backbone pretrained on the Kinetics dataset [8] to obtain features $\mathbf{X_i} \in \mathbb{R}^{1024 \times l_i}$ from the global pooling layer.

R(2+1)D: For the R(2+1)D [15] network, we re-scale frame pixels between 0 and 1, after which we resize all frames to 128×171. We then normalize these frames and fi-

[1] https://pytorch.org/vision/0.8/models.html
[2] https://github.com/deepmind/kinetics-i3d

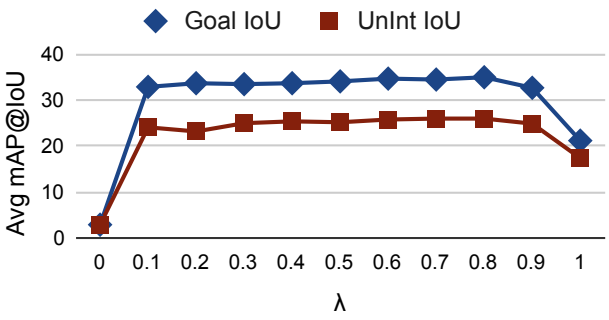

Figure 4. Analysis of the effect of $lambda$ which is a scalar parameter for controlling the tradeoff between the MIL Loss and Overlap Regularization.

nally apply center crop to obtain 112×112 frames. We the chunk the frames in the same way and pass it through the R(2+1)D [15] backbone pretrained on Kinetics to obtain features $\mathbf{X_i} \in \mathbb{R}^{512 \times l_i}$ from the global pooling layer.

8. Annotation Tool for Label Evaluation and Correction

The annotation tool used for the human evaluation and correction process is shown in Fig.5. We provide a video to the evaluator along with the actions extracted from the annotations. The evaluator can then view the videos and mark the goal-directed actions as well as unintentional action as either 'Good' (G) or 'Poor' (P), with reference to the video. 'Good' is given to an action which is entailed in the video and 'Poor' otherwise. In case the evaluator marks an action as 'Poor', they can then choose another action from the already present list of total actions, or else add a new action if not contained in the list. The evaluator also has an option to not keep the video in the case the goal of the agent in the video was ambiguous. Once this process is complete, evaluators can hit 'Submit', which would then load the next video.

9. Qualitative Results of Goal-directed and Unintent. Action Localization

In this section, we provide additional qualitative results of our model, along with previous weakly supervised action localization (WSAL) models, namley WTALC [13] and STPN [12]. We have provided examples of videos containing diverse actions, in order to show our model's generalizability. From Fig. 6, Fig. 7 and Fig. 8, we notice that our model is able to focus on distinct regions in order to infer the goal-directed and unintentional actions, whereas the previous WSAL models focus on overlapping regions, and in many cases have very sparse attention weights. We conjecture this is due to the nature of task these models were

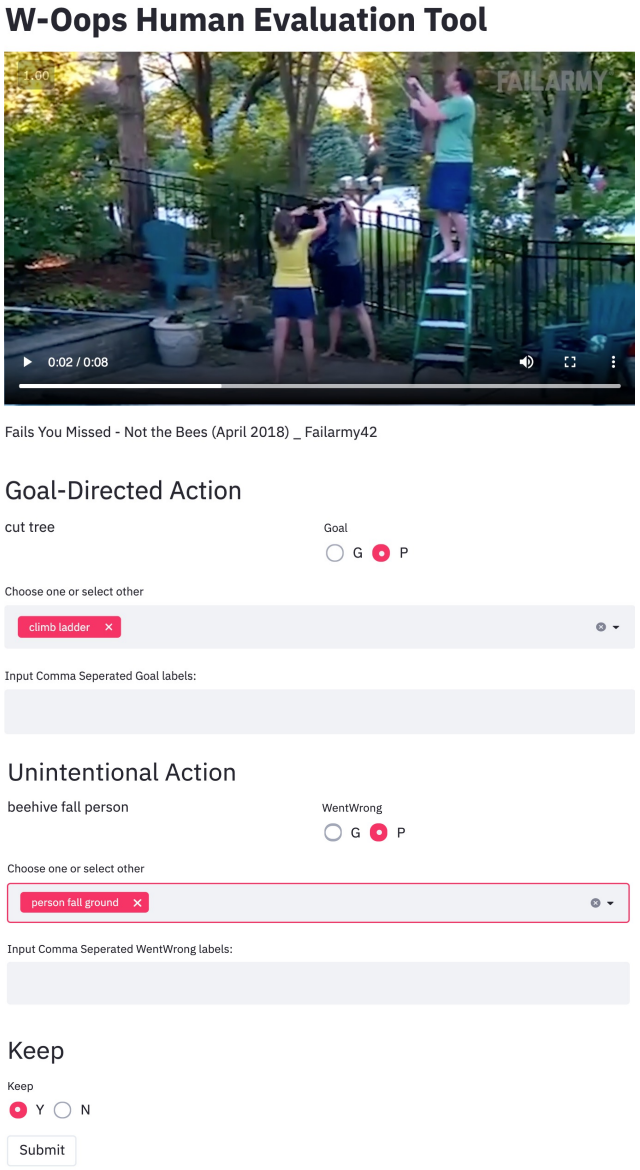

Figure 5. Interface for W-Oops annotations, where we ask the annotators to rate the semi-automatically extracted goal-directed and unintentional actions as 'Good' or 'Poor'. If 'Poor', they can choose from a fixed list of already present actions or input their own. They also have an option to indicate whether or not to keep the video in the case the goal in the video is ambiguous.

originally built for, *i.e.*, segmenting atomic actions from untrimmed videos. Additionally, we can see that the Overlap Regularization is able to enforce our model to maintain the temporal ordering of the goal-directed/unintentional action.

10. Qualitative Results for Video Captioning

This section provides qualitative results of the video captioning experiment. We report the ground-truth captions annotated by humans, captions generated without using our localization module, as well as captions generated using our localization module. Fig. 9 shows that leveraging our localization module helps generate more descriptive and semantically correct captions, being able to describe the video better and hence assisting in the teleological understanding.

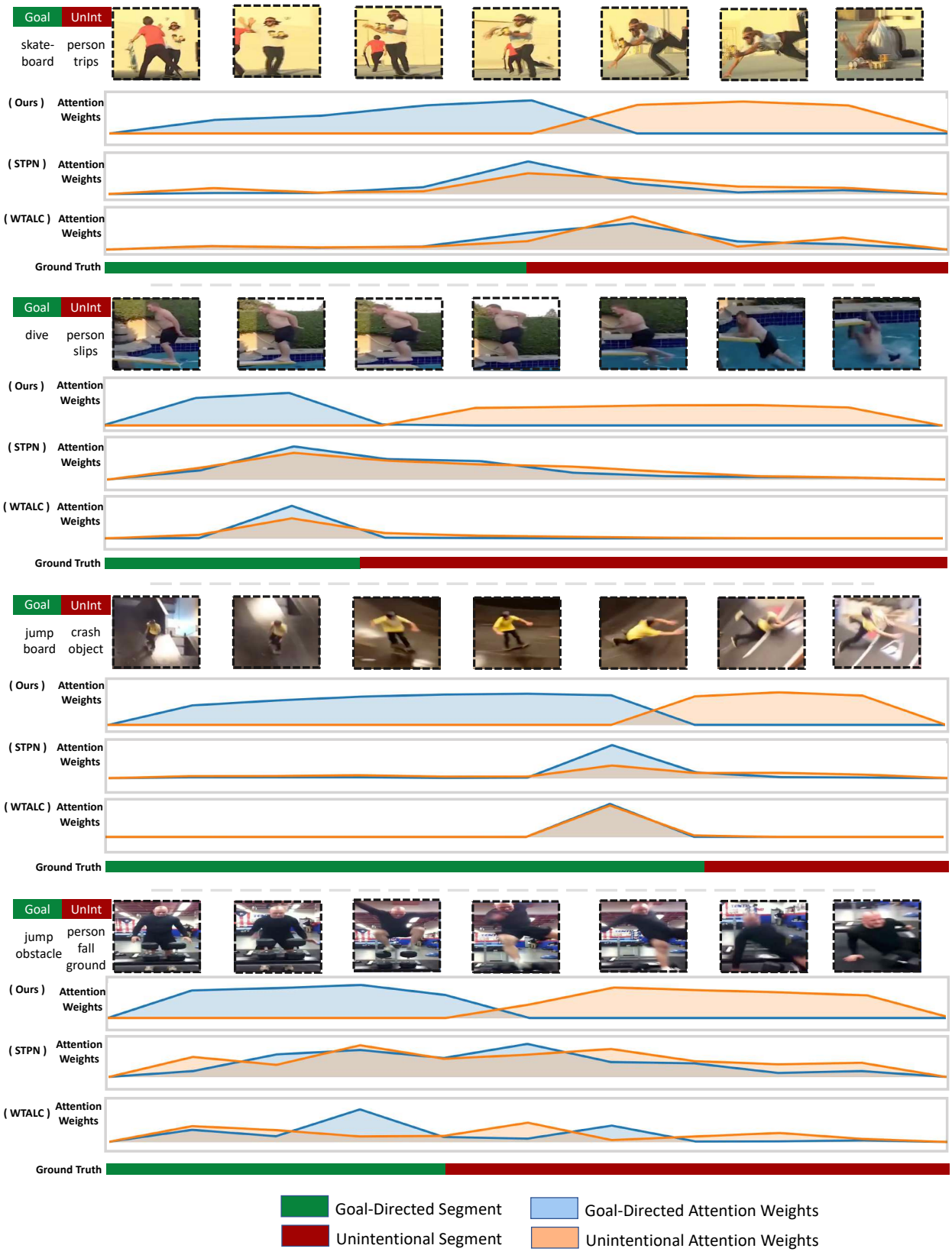

Figure 6. Qualitative results of our model's outputs. We provide attention weights outputted from STPN trained on our dataset, as well as the ground truth segments for comparison.

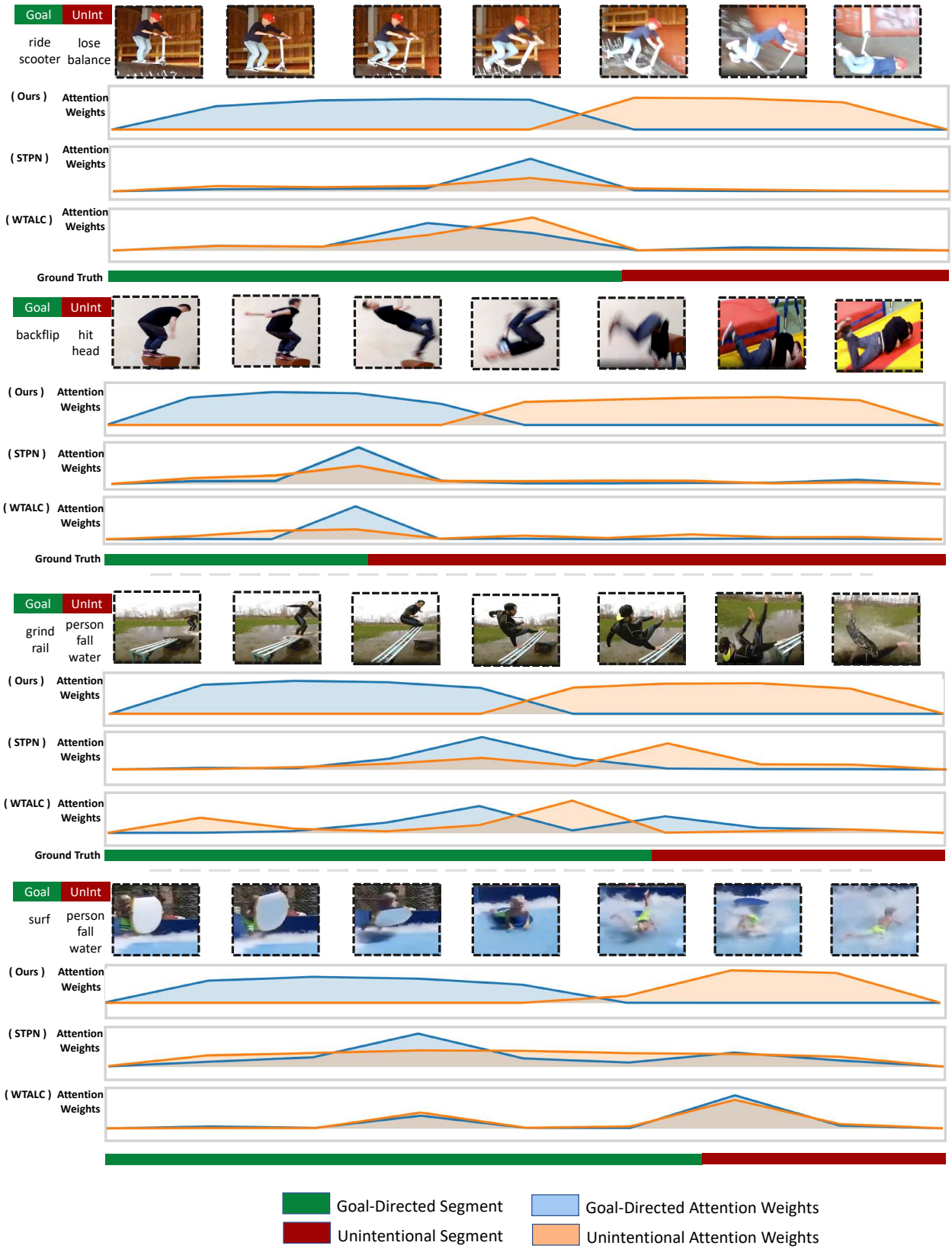

Figure 7. Qualitative results of our model's outputs. We provide attention weights outputted from STPN trained on our dataset, as well as the ground truth segments for comparison.

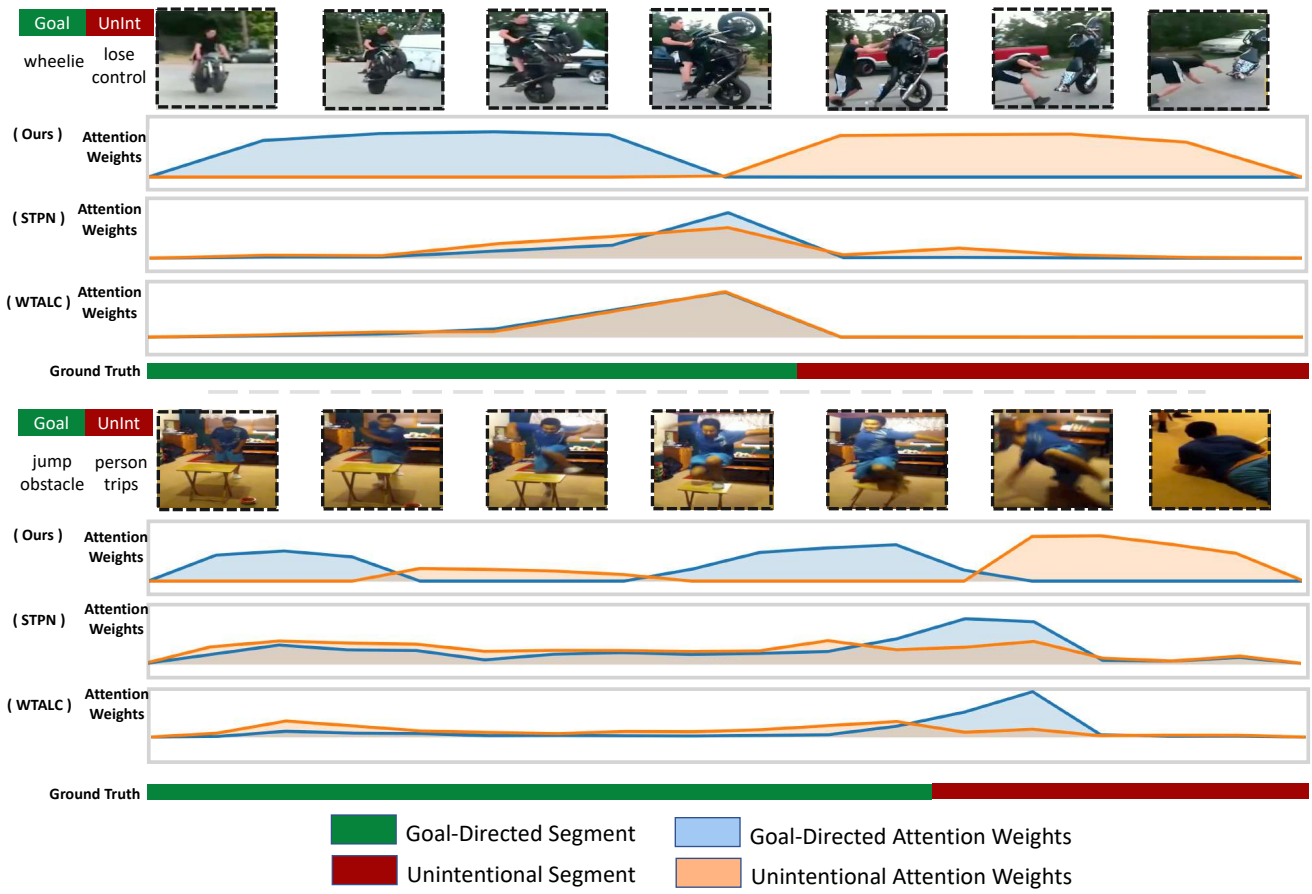

Figure 8. Qualitative results of our model's outputs. We provide attention weights outputted from STPN trained on our dataset, as well as the ground truth segments for comparison.

GT: Man try to perform skateboard trick but he ended up face planting on to the ground
Without Loc: A man try to do a flip on a flip but he fall off the ground and fall on the ground
With Loc: A man is trying to jump on a skateboard but he fell off the board and fall on the ground

GT: Man attempted to walk on a balance beam over a pool but man lost his balance and fell into the pool
Without Loc: A man is trying to jump on a pool but he fall into the pool and fall into the water
With Loc: A man is trying to jump off a pool but the man loses his balance and fall into the water

GT: The kid wants to jump off the table but the kid lost his balance fall on the floor and cry
Without Loc: A man is trying to jump a backflip but he fall off the ground and fall on the ground
With Loc: A man is trying to jump on a <unk> but he hit his head and he fell on his face

GT: a full family was attempting to jump on a trampoline but the father jumped hard into the trampoline and it ripped below them
Without Loc: a man is trying to jump on a bar but the man fall on the bar and fall on the ground
With Loc: A man is trying to jump on a trampoline but the man slipped and fall on the ground

Goal-Directed Description Unintentional Description

Figure 9. Qualitative results for the video captioning experiment. We provide ground truth captions from a human annotator, captions generated without as well as with our localization module. We observe that the captions generated leveraging our localization module tend to be more descriptive and semantically correct.

References

[1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Minjie Hua, Yibing Nan, and Shiguo Lian. Falls prediction based on body keypoints and seq2seq architecture. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[8] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[9] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[10] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[11] Daisuke Miki, Shi Chen, and Kazuyuki Demachi. Weakly supervised graph convolutional neural network for human action localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 653–661, 2020.

[12] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.

[13] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[14] Markus D Solbach and John K Tsotsos. Vision-based fallen person detection for the elderly. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1433–1442, 2017.

[15] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[17] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 915–922, 2013.

[18] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[19] Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*, 2020.

[20] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[21] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[22] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020.

[23] Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. Feature pyramid transformer. In *European Conference on Computer Vision*, pages 323–339. Springer, 2020.

[24] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288, 2020.

[25] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.