Video Representations of Goals
Emerge from Watching Failure

Dave Epstein and Carl Vondrick
Columbia University
aha.cs.columbia.edu

Abstract

We introduce a video representation learning framework that models the latent goals behind observable human action. Motivated by how children learn to reason about goals and intentions by experiencing failure, we leverage unconstrained video of unintentional action to learn video representations without direct supervision. Our approach models videos as contextual trajectories that represent both low-level motion and high-level action features. Experiments and visualizations show the model is able to predict underlying goals and detect when action switches from intentional to unintentional. Although the model is trained with minimal supervision, it is competitive with highly-supervised baselines, showing the role of failure examples for learning goal-oriented video representations.

1 Introduction

Consider the person in Figure 1, which shows a man heating a wine bottle with a blowtorch. Even if this action is unconventional, we still perceive the action as rational in the context of the goal (to open the bottle). Evidence suggests that this ability to reason about goals is learned before our second birthday [40, 47], and it plays a key role in children’s rapid development of communicative skills [41] and mental representations of the world [2]. However, despite the importance of this problem, learning visual representations of human goals has remained challenging.

Visual action recognition has largely focused on learning to recognize action categories [3, 19], which indicate how a person acted, and not why they acted. While this has spurred tremendous progress in video analysis, the resulting video representations do not discriminate the underlying goals of action. We hypothesize that a key missing piece is the lack of examples demonstrating the failure to achieve goals. Similar to how a child learns about goals by experiencing failure, we leverage a large dataset containing both intentional and unintentional real-world action [6] to learn goal-oriented representations of video.

Figure 1: What is this person’s goal? Although only the action is observable, we are still able to predict the goal behind the action (to open the bottle). In this paper, we introduce a model to learn video representations that encode goals as latent action trajectories.
We present a video model that learns a trajectory representation of action, and encodes goals as the path of the trajectory. We input entire videos to our model by first dividing them into short clips, which are run through a 3D CNN to learn low-level motion features. We then pass the motion features into a Transformer model, which models relations between different periods in its input, and thus represents the entire action as context-aware latent trajectories. The whole model is trained from scratch in an end-to-end manner.

Our experiments show that observing failure is vital for learning representations of goals. We evaluate our model on three visual tasks for goal prediction. First, we experiment on detecting unintentional action in video, and we demonstrate state-of-the-art performance on this task. Second, we evaluate the representation at predicting goals with minimal supervision, which we characterize as structured categories consisting of subject, action, and object triplets. Lastly, we use our representation to automatically “correct” unintentional action and decode these corrections by retrieving from other videos or generating categorical descriptions.

Our main contribution is an approach that (1) models long video sequences as latent-space trajectories with indirect supervision and, in doing so, (2) learns a goal-directed representation of videos. Since the goals are encoded in the path of the trajectory, we also show how to find minimal adjustments to the path to automatically correct unintentional action in video. The remainder of this paper will describe this approach in detail. Code, data, and models are available online.

2 Related Work

Recognizing action in video: Previous work explores many different approaches to recognizing action in video. Earlier directions develop hand-designed features to process spatio-temporal information for action recognition [24, 22, 44, 33]. Popular deep learning architectures for images were...
Figure 3: Labeling goals and failures in video: To evaluate our representation, we annotate the Oops! dataset with short sentences describing the goals and failures. We extract subject-verb-object triples and train a decoder on learned representations. The intentional and unintentional action in the dataset span a diverse range of categories.

extended to operate directly on video by modeling time as a third dimension [13, 3, 36, 26, 19]. To deal with variable-length or long video input, previous work frequently takes one of two approaches: pooling or recurrent networks. However, pooling loses spatial and/or temporal connections between different moments of video. Since recurrent networks are sequential, they require selecting important video features ahead of time, without viewing full context. RNNs are also known to struggle to connect between far-apart inputs, which creates significant challenges in modeling long-term video.

[39] is most similar to our approach, since they also run clips through 3D CNNs and Transformers, but they freeze 3D CNNs and train on a “masked video modeling” task, ultimately discarding contextually learned temporal dynamics across videos since their goal is to learn information useful for an effective cross-modal representation. To address these drawbacks, we propose a 3D-CNN-Transformer model which allows for short-term, granular motion detection combined with a long-term action representation, trained end-to-end from scratch.

**Learning about intention:** Evidence in developmental psychology quantifies why humans perceive intention [2], how we perceive it [48, 49, 47], when we begin to do so [27, 28], and what allows us to infer the goals behind others’ behavior [35]. While these questions have been studied in early stages of child development, the same abilities have remained a challenge for machines in unconstrained situations. One possible reason for this is a lack of realistic data. We take advantage of incidental signals in unconstrained videos [6] to learn video representations.

**Leveraging adversarial attacks:** We use adversarial gradients [10, 23] to find corrections to the trajectory. Previous work studied adversarial attacks in steganography [14, 52], software bug-finding [34], generating CAPTCHAs [43] to fool modern deep nets [32], generating interesting images [37], creating real-world 3D objects that trick neural networks [51, 1], and in training models more robust to test-time adversarial attacks [30, 10, 31]. [17] extend this concept to generative models, setting a new image output as a target label and perturbing latent space. In video, [21, 46] introduce various methods to fool action recognition networks, often on a 3D CNN backbone. We instead utilize adversarial attacks to manipulate and correct unintentional action.

3 **Method**

In this section, we introduce our framework to learn video representations as trajectories, formulate learning objectives, and use the learned representations to predict goals in video.

3.1 **Visual Dynamics as Trajectories**

The conventional approach to representing video data is to run each clip through a convolutional network and combine clip representations by pooling to run models on entire sequences [7, 12, 9, 50]. However, these methods do not allow for connections between different moments in video and cannot richly capture temporal relationships, which give rise to goal-directed action. While recurrent networks [15] are more expressive, they require compressing history into a fixed-length vector, which
forces models to select relevant visual features without viewing full context and makes reasoning about connections between different moments difficult, especially when they are far apart.

Temporal streams of visual input are highly contextual with both short- and long-term dependencies. We will represent video as a contextually-adjusted trajectory of latent representations in a learned space. Figure 2 illustrates this architecture, which has both a motion and action level:

**Motion Level:** First, we separate video into short clips (or tokens) in order to make initial motion-level observations. Let $x$ be a video, and $x_i$ be a video clip centered at time $i$. We estimate the motion-level features $\phi_i = f(x_i)$ where $f$ is a 3D CNN [20].

**Action Level:** Second, we model relations between $\phi_i$ to construct a contextual trajectory $h_i = g_i(\phi)$ where $g$ is the Transformer [42]. The Transformer architecture is able to capture relations in its input by performing self-attention among tokens in its input sequence, and outputs a contextual representation across the video. Since the Transformer architecture can incorporate contributions from both nearby and far away moments in its representations for each clip, it is well-suited to modeling higher-level connections between the atomic actions recognized at the motion level. The resulting sequence of embeddings induces a trajectory in the form of a sequence of hidden vectors $h_i$, which we can use for different downstream tasks.

### 3.2 Learning with Indirect Supervision

We train the representation with indirect supervision that is accessible at large scales. We use the following two objectives for learning:

**Action Intentionality:** We train the model to temporally localize when action is unintentional. Let $t \in \mathbb{R}$ be the video frame where the action shifts from intentional to unintentional (which we assume is labeled [6]). For each video clip $x_i$, we set the target $y_{i}^{\text{fail}} \in \{0, 1, 2\}$ according to whether the labeled $t$ happens before, during, or after the clip $x_i$. The model estimates $\hat{y}_{i}^{\text{fail}} = \text{softmax}(w_1^T h_i)$ with a linear projection where $w_1$ is a jointly learned projection matrix to $\mathbb{R}^3$. We train with a cross-entropy loss between $y_{i}^{\text{fail}}$ and $\hat{y}_{i}^{\text{fail}}$ where the class weight is set to the inverse frequency of the class label to balance training. We label this loss $L_{\text{fail}}$.

**Temporal Consistency:** We also train the model to learn temporal dynamics with a self-supervised consistency loss [12, 29, 8, 45, 18, 5]. Let $y_{\text{mp}} = 1$ indicate that the sequence is consistent. We predict whether the input sequence is temporally consistent with $\hat{y}_{\text{mp}} = \sigma(w_2^T h_0)$ where $w_2$ is a jointly learned projection to $\mathbb{R}$. We train with the binary cross-entropy loss between $y_{\text{mp}}$ and $\hat{y}_{\text{mp}}$. We label this loss $L_{\text{mp}}$ (next sequence prediction).

We create inconsistent sequences as follows: For each video sequence in the batch, we bisect the sequence into two parts at a random index with probability $p_{\text{split}} = 0.5$. For these sequences, we perturb one or both of the video segments with probability $p_{\text{perturb}} = 0.5$. When perturbing, we swap the order of the two sequences with probability $p_{\text{swap}} = 0.3$, otherwise we pick a randomly sized subsequence from another video sequence in the batch to replace one of the two segments.

**Training:** To train our model, we set the overall loss as $L = L_{\text{fail}} + \lambda L_{\text{mp}}$, where $\lambda$ is a hyperparameter controlling the importance of the coherence loss. We set $\lambda = 0.5$ to balance the magnitudes of the losses. We sample sequences of one-second long clips, run each clip $x_i$ through the motion-level 3D

---

**Figure 4:** **Automatically correcting unintentional action:** Starting from an initial trajectory, we use model gradients as a signal to correct the course of points representing unintentional action. This corrected trajectory is evaluated by decoding into various feature spaces.
### Localization

| Method | Localization 0.25 sec | Localization 1 sec | Classification Accuracy |
|--------|------------------------|--------------------|-------------------------|
| Kinetics supervision [3] | 69.2 | 37.8 | 53.6 |
| Kinetics supervision [3] + finetune | 75.9 | 46.7 | 64.0 |

| Method | Classification only | Kinetics supervision [3] + finetune |
|--------|---------------------|-----------------------------------|
| 3D CNN only [6] | 68.7 | 39.8 | 59.4 |

| Method | Classification only + Pseudo-GT | Kinetics supervision [3] + finetune + Pseudo-GT |
|--------|---------------------------------|-----------------------------------------------|
| Our model | 64.9 | 33.6 | 73.0 |

| Method | Classification only + Coherence loss | Kinetics supervision [3] + finetune + Coherence loss |
|--------|-------------------------------------|-----------------------------------------------|
| Our model | 71.8 | 39.6 | 77.8 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 72.4 | 39.9 | 77.7 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 71.8 | 39.6 | 77.8 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 72.4 | 39.9 | 77.7 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 71.8 | 39.6 | 77.8 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 72.4 | 39.9 | 77.7 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 71.8 | 39.6 | 77.8 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 72.4 | 39.9 | 77.7 |

| Method | Classification only + Pseudo-GT + Coherence loss | Kinetics supervision [3] + finetune + Pseudo-GT + Coherence loss |
|--------|--------------------------------------------------|---------------------------------------------------------------|
| Our model | 71.8 | 39.6 | 77.8 |

### Table 1: Detecting unintentional action

We evaluate models on classifying and localizing unintentional action. Our model is competitive with Kinetics pretraining despite training from scratch, and outperforms it on classification.

CNNs then pass all outputs through the Transformer stack, and calculate the gradients. We optimize the loss with stochastic gradient descent. At inference time, we run entire continuously-sampled videos through our model.

### 3.3 Completing Goals by Auto-Correcting Trajectories

We use this learned representation in order to complete the goals of people in the scene [27, 38]. However, since the model is trained with indirect supervision, the trajectories $h$ are not supervised with goal states. We propose to formulate goal completion as a latent trajectory prediction problem. Given an observed trajectory of unintentional action $h$, we seek to find a new, minimally modified trajectory $h'$ that is classified as intentional. By analogy to how word processors auto-correct a sentence, we call this process action auto-correct. We illustrate this process in Figure 4.

We find this correction in feature space, not pixel space, to yield interpretable results. We find a gradient to the features $\phi$ that switches the prediction $\hat{y}_i = 0$ to be the “intentional” category for all clips $i$. We formulate an optimization problem with two soft constraints. Firstly, we want to increase the classification score of intentional action $L_{\text{fail}}$. Secondly, we want the resulting trajectory to be temporally consistent $L_{\text{nsp}}$. Without this term, the corrected trajectory is not required to be coherent with the initial part of the original trajectory. We minimize:

$$
\min_{\phi'} J \quad \text{where} \quad J = \max (0, L_{\text{nsp}}(\phi') - L_{\text{nsp}}(\phi)) + \lambda \sum_i L_{\text{fail}}(\phi')
$$

where $L$s are the original loss functions but with target labels $y_{\text{fail}}$ overrodden to be the intentional class, and $\lambda = 2$ is a scalar to balance the two terms. We only modify $\phi$ on the clips which the model classifies as unintentional in the first place, which we denote $\phi'_{i,T}$. The coherence loss is also truncated by its original value, causing the optimization to favor a trajectory that is no less temporally coherent than the original one.

To solve this optimization problem, we use the iterative target class method [23], which repeatedly runs the input through the model and modifies it in the direction of the desired loss. For every $\phi_i$ corresponding to a clip where action is unintentional, we repeat a gradient attack step towards the target $y_{\text{fail}}^i = 0$. The complete update is:

$$
\phi_{k+1}^{i,T} = \text{clip} \left[ \phi_{k,T}^{i} - \alpha \text{sign} (\nabla \phi_{i,T} J), \phi_{i,T} + \epsilon \right]
$$

where $\phi_{0,T} = \phi_{i,T}$. We repeat this process until the network is “fooled” into classifying the input as intentional action, for at most $k_{\text{max}}$ iterations or until $\arg \max y_{\text{fail}}^i = 0$. Once the halting condition is satisfied, we run the modified $\phi'$ vectors through the model, yielding a trajectory of corrected action $h'$ that encodes successful completion of the goal. This trajectory can be read out into various spaces (Section 5.1).

In other words, goals are the adversarial examples [10] of failed action – instead of viewing adversarial examples as a bug, we view them as a feature [16].
Decoding the Trajectories: After estimating the decoder, we read out triplets from different parts of videos. The first row shows intentional action, and the decoder predicts the goal. The second row shows unintentional action, and the decoder now predicts the failure instead. The final row shows unintentional videos that have been auto-corrected, and the decoder returns to predicting goals, suggesting the auto-correct procedure shifts the failed trajectories towards successful ones.

4 Unintentional Action and Goals Dataset

Similar to how children learn about goals by perceiving failed attempts at executing them [28], we hypothesize that examples of failure are crucial for learning to discriminate between action and goal. We use the recently released Oops! dataset [6], which is a large collection of videos containing intentional and unintentional action, to train and evaluate our models. Videos in this dataset are annotated with the moment at which action becomes unintentional. Figure 3 shows some example frames. We also use the Kinetics dataset [4] to evaluate models, since it contains a wide range of successful action.

Goal Annotation: Established action datasets in computer vision [11, 25] contain annotations about person and object relationships in scenes, but they do not directly annotate the goal, which we need for evaluation of goal prediction. We collect unconstrained natural language descriptions of a subset of videos in the Oops! dataset (4675 training videos and 3404 test videos), prompting Amazon Mechanical Turk workers to answer “What was the goal in this video?” as well as “What went wrong?”. We then process these sentences to detect lemmatized subject-verb-object triples, manually correcting for common constructions such as “tries to X” (where the verb lemma is detected as “try”, but we would like “X”). The final vocabulary contains 3615 tokens. Figure 3 shows some example annotations. We use SVO triples to evaluate the video representations.

5 Experiments

We experiment with our model on two tasks: recognizing intentional action, and predicting goals. We train our method from scratch on a dataset of unintentional action [6].

---

1We found $k_{max} = 25$, $\alpha = 0.03$, $\epsilon = 1$ to be reasonable values.
2In addition to the ground truth annotations provided by [6], we run their pretrained model on the unlabeled portion of the training set and collect pseudo-ground-truth, which we found improves performance.
3with > 10k approvals at a ≥ 99% rate
4Using the Spacy.io natural language library
Figure 6: **Retrievals from Auto-corrected Trajectories:** We show the nearest neighbors from auto-corrected action trajectories, using our proposed method and a linearization baseline. The retrievals are computed across both the Oops! and Kinetics datasets. The corrected representations yield corrected trajectories that are often embedded close to the goal.

### Table 2: Comparison of Representations

To evaluate how well representations encode goals, we freeze them and estimate a linear projection to predict labelled subject-verb-object triples.

| Method         | Subject Top 1 | Subject Top 5 | Verb Top 1 | Verb Top 5 | Object Top 1 | Object Top 5 | Average Top 1 | Average Top 5 | All three Top 1 | All three Top 5 |
|----------------|---------------|---------------|------------|------------|--------------|--------------|---------------|--------------|----------------|----------------|
| Kinetics [3]   | 26.79         | 72.34         | 27.33      | 52.67      | 36.01        | 64.64        | 30.04         | 63.22        | 2.07           | 16.46          |
| 3D CNN [6]     | 29.44         | 72.72         | 26.42      | 50.36      | 44.71        | 57.89        | 33.52         | 60.32        | 2.86           | 13.85          |
| Scratch        | 23.67         | 55.73         | 22.74      | 45.44      | 44.82        | 52.67        | 30.41         | 51.28        | 1.42           | 8.72           |
| Our Model      | **34.31**     | **74.50**     | **29.72**  | **54.17**  | **44.95**    | **62.15**    | **36.32**     | **62.27**    | **3.32**       | **14.39**      |
| Chance         | 0.14          | <0.01         |            |            |              |              |               |              |                |                |

#### 5.1 Experimental Setup

**Baselines:** We evaluate the 3D CNN from [6] which is trained from scratch on the action intentionality loss (Section 3.2). We also evaluate a 3D CNN pre-trained on Kinetics action recognition, which is frozen unless indicated otherwise. We compare goal prediction to a frozen, randomly initialized network (denoted “Scratch”). We also consider several ablations of our model. To evaluate representations, we freeze them and implement different decoders as described below.

**Retrieval:** This decoder does not require further training and performs nearest-neighbor retrieval among one-second long clips in the test sets for the Oops! and Kinetics datasets. While we do not learn a representation using Kinetics, we include it in retrieval to see if auto-corrected actions match with successfully executed goals in Kinetics rather than failed attempts (see Section 5.3). This decoder maintains a lookup table of all clip representations and computes the $k$-nearest neighbors from different videos using cosine distance.

**Categorization:** We also implement a decoder using the textual labels we gathered on the videos. Here, the task is to describe the goals of the input video using the SVO triplets. We train a decoder to predict the main goal for clips with intentional action, and predict what went wrong for clips with unintentional action. The estimated decoder will describe the video with descriptions of the goal, for example “athlete wins game”, which is a goal, and not “woman throws ball”, which is an action. We train a linear layer to output a vector for subject, verb, and object. As ground truth, we use BERT word embeddings [5], calculating scores using dot product and running them through softmax and a cross-entropy loss.

#### 5.2 Unintentional Action Detection

We first evaluate how well the model is able to detect and localize when action deviates from its goal. We use labels from the test set in [6] as the ground truth. We process entire videos with our model, sampling continuous one-second clips as tokens, and take the predicted localization as the center of the clip with maximum probability of failure. We also classify each clip according to its label (intentional, transitional, or unintentional). We show results in Table 1. On the former task, our model is competitive with fine-tuning a fully-supervised Kinetics CNN, despite using less data and
Figure 7: Analyzing the Representation: We probe the learned trajectories. (a) shows the neurons with highest correlation to the words in the SVO vocabulary, along with their top-5 retrieved clips. Neurons that detect intentions across a wide range of action and scene appear to emerge, despite only training with binary labels on the intentionality of action. (b) We show six randomly sampled video trajectories in t-SNE space, before and after auto-correct, superimposed over the embeddings for intentional and unintentional action. Visualizations suggest our approach tends to adjust unintentional action in the direction of successful, intentional action.

Table 3: Evaluating Autocorrection: We show effect of auto-correct on SVO decoder predictions (top 5 accuracy and rank assigned to the correct triple). Our model shifts probability mass from unintentional to intentional SVOs.

| Method             | Intentional SVO |       | Unintentional SVO |       |
|--------------------|-----------------|-------|-------------------|-------|
|                    | Δ Acc.          | Δ Rank| Δ Acc.            | Δ Rank|
| Kinetics [3]       | +0.4            | +0.3M | -0.3              | -1.2M |
| 3D CNN [6]         | +0.3            | +0.1M | -0.3              | -0.6M |
| Ours (linearized)  | +1.6            | +15.8M| -3.3              | -9.3M |
| Ours (adversarial) | +1.6            | +15.8M| -3.3              | -9.3M |

less supervision. On classification, our network outperforms the Kinetics network by 14%, showing that representing videos as contextual trajectories is effective.

5.3 Goal Prediction

We next evaluate the model at predicting goal descriptions. We train a decoder on the trajectory to read out subject, verb, object triplets. In training, if sentences have more than one extracted SVO, we randomly select one as ground truth. In testing, we average-pool predictions among all clips with intentional action and unintentional action separately and take the maximum over all sentence SVOs. Each video clip has two pooled predictions: one for intentional action, and one for unintentional action. Table 2 shows our model obtains better top-1 accuracy on all metrics than baselines, including the Kinetics-pretrained model, and is competitive on top-5 accuracy.

5.4 Analysis of Learned Representation

To evaluate how action and goals are embedded in the trajectory representation, we find the minimal “auto-correction” to the unintentional action sequences and probe them. As a comparison, we implement a simple baseline where we linearly extrapolate the trajectory of observed intentional action: if the unintentional action in a sequence of clips \( \{x_i\}_{i=0}^n \) begins at clip \( j \), we extend the trajectory for a clip \( x_k \in \{x_j, \ldots, x_n\} \) by setting \( h_k = h_j + (k - j) \frac{h_n - h_0}{j} \).

Figure 6 shows examples of nearest neighbor retrievals of the corrected latent vectors, computing over the Oops! and Kinetics test sets. Despite not training on Kinetics (i.e. on videos with completed goals), our representation can adjust video trajectories such that their nearest neighbors are goals being successfully executed. We also examine the effects of auto-correction on the frozen SVO decoder. Table 3 shows these results. For decoders trained on all models, rankings of intentional action SVOs increase while those of unintentional SVOs decrease. However, the changes are greatest for our model. Figure 5 visualizes the output of a frozen SVO decoder on auto-corrected actions, demonstrating the auto-correct process’ ability to encode completed goals in its output trajectories.
We finally probe the model’s learned representation to analyze how trajectories are encoded. We measure Spearman’s rho correlation between the activation of neurons in the output vectors $h \in \mathbb{R}^{512}$ and words in the SVO vocabulary. Each video is an observation containing neuron activations and an indicator variable for whether each word is present in ground truth. Many neurons have significant correlation, and we show the top 3 in Figure 7a, along with the 5 clips that activate them most. These neurons appear to discover common actions in the Oops! dataset, despite being trained without any action labels. We also visualize trajectories of some videos using t-SNE (Figure 7b), before and after autocorrect. Our model often adjusts trajectories from unintentional action to the region of embedding space with Kinetics videos, shown in the figure as “at goal” action.

6 Conclusion

We introduce an approach to represent videos as contextual trajectories in a learned latent space, leveraging the Transformer architecture. By encoding action as a trajectory, we are able to perform several different tasks, such as decoding to categorical descriptions or manipulating the trajectory. Our experiments show that learning from failure examples, not just successful action, is crucial for learning rich visual representations of goals.

Acknowledgements

We thank Didac Surís, Mia Chiquier, Amogh Gupta, Ruoshi Liu, Ishaan Chandratreya, and Boyuan Chen for helpful comments. Funding was provided by DARPA MCS, NSF NRI 1925157, and an Amazon Research Gift. We thank NVidia for donating GPUs.

Broader Impact

Human action recognition is critical for situational awareness applications in robotics, healthcare, and security, which may potentially have a large practical impact on society. For example, predicting the goals of actions could enable machines to better assist and communicate with people. A key limitation in our experiments is that we leverage publicly available video data, which is likely biased to Western cultures. Consequently, the learned representation likely encodes a Western definition of success, which may not generalize to other demographic areas.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017. 3

[2] John Barresi and Chris Moore. Intentional relations and social understanding. Behavioral and brain sciences, 19(1):107–122, 1996. 1, 3

[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. 1, 3, 5, 7, 8

[4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. arXiv preprint arXiv:1808.01340, 2018. 6

[5] 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. 4, 7

[6] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. arXiv preprint arXiv:1911.11206, 2019. 1, 3, 4, 5, 6, 7, 8

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 6202–6211, 2019. 3
[8] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. 4

[9] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. *arXiv preprint arXiv:1912.04487*, 2019. 3

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 3, 5

[11] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018. 6

[12] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3, 4

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 3

[14] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*, pages 1954–1963, 2017. 3

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. 3

[16] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 125–136. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8307-adversarial-examples-are-not-bugs-they-are-features.pdf. 5

[17] Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 3

[18] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016. 4

[19] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. *arXiv preprint arXiv:1912.06992*, 2019. 1, 3

[20] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231, 2012. 4

[21] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019. 3

[22] Alexander Klaser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. 2008. 2

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 3, 5

[24] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3): 107–123, 2005. 2
[25] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020. 6

[26] 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. 3

[27] Andrew N Meltzoff. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental psychology*, 31(5):838, 1995. 3, 5

[28] Andrew N Meltzoff, Alison Gopnik, and Betty M Repacholi. Toddlers’ understanding of intentions, desires and emotions: Explorations of the dark ages. 1999. 3, 6

[29] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 4

[30] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015. 3

[31] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016. 3

[32] Margarita Osadchy, Julio Hernandez-Castro, Stuart Gibson, Orr Dunkelman, and Daniel Pérez-Cabo. No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation. *IEEE Transactions on Information Forensics and Security*, 12(11):2640–2653, 2017. 3

[33] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 612–619, 2014. 2

[34] Dongdong She, Kexin Pei, Dave Epstein, Junfeng Yang, Baishakhi Ray, and Suman Jana. Neuzz: Efficient fuzzing with neural program smoothing. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 803–817. IEEE, 2019. 3

[35] Thomas R Shultz, Diane Wells, and Mario Sarda. Development of the ability to distinguish intended actions from mistakes, reflexes, and passive movements. *British Journal of Social and Clinical Psychology*, 19(4):301–310, 1980. 3

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 3

[37] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 3

[38] Alexander Skulmowski, Andreas Bunge, Bret R Cohen, Barbara AK Kreilkamp, and Nicole Troxler. Investigating conceptions of intentional action by analyzing participant generated scenarios. *Frontiers in psychology*, 6:1630, 2015. 5

[39] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019. 3

[40] Michael Tomasello. The usage-based theory of language acquisition. In *The Cambridge handbook of child language*, pages 69–87. Cambridge Univ. Press, 2009. 1

[41] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrikre Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(3):675–691, 2005. 1
[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017. 4

[43] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 294–311. Springer, 2003. 3

[44] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In CVPR 2011, pages 3169–3176. IEEE, 2011. 2

[45] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8052–8060, 2018. 4

[46] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641, 2018. 3

[47] Amanda L Woodward. Infants’ grasp of others’ intentions. Current directions in psychological science, 18(1):53–57, 2009. 1, 3

[48] Amanda L Woodward, Jessica A Sommerville, and Jose J Guajardo. How infants make sense of intentional action. Intentions and intentionality: Foundations of social cognition, pages 149–169, 2001. 3

[49] Amanda L Woodward, Jessica A Sommerville, Sarah Gerson, Annette ME Henderson, and Jennifer Buresh. The emergence of intention attribution in infancy. Psychology of learning and motivation, 51:187–222, 2009. 3

[50] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In Proceedings of the IEEE international conference on computer vision, pages 5783–5792, 2017. 3

[51] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. Invisible mask: Practical attacks on face recognition with infrared. arXiv preprint arXiv:1803.04683, 2018. 3

[52] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 657–672, 2018. 3