We Have So Much in Common: Modeling Semantic Relational Set Abstractions in Videos

Alex Andonian[1](✉), Camilo Fosco[1], Mathew Monfort[1], Allen Lee[1],
Rogerio Feris[2], Carl Vondrick[3], and Aude Oliva[1]

[1] Massachusetts Institute of Technology, Cambridge, USA
{andonian,camilolu,mmonfort,allenlee,oliva}@mit.edu
[2] MIT-IBM Watson AI Lab, Cambridge, USA
rsferis@us.ibm.com
[3] Columbia University, New York, USA
vondrick@cs.columbia.edu

Abstract. Identifying common patterns among events is a key capability for human and machine perception, as it underlies intelligent decision making. Here, we propose an approach for learning *semantic relational set abstractions* on videos, inspired by human learning. Our model combines visual features as input with natural language supervision to generate high-level representations of similarities across a set of videos. This allows our model to perform cognitive tasks such as *set abstraction* (which general concept is in common among a set of videos?), *set completion* (which new video goes well with the set?), and *odd one out detection* (which video does not belong to the set?). Experiments on two video benchmarks, Kinetics and Multi-Moments in Time, show that robust and versatile representations emerge when learning to recognize commonalities among sets. We compare our model to several baseline algorithms and show that significant improvements result from explicitly learning relational abstractions with semantic supervision. Code and models are available online (Project website: abstraction.csail.mit.edu).

Keywords: Set abstraction · Video understanding · Relational learning

1 Introduction

Humans are extraordinary at picking out patterns between different events, detecting what they have in common and organize them into abstract categories, a key ability for everyday reasoning. Our goal in this paper is to instantiate this

A. Andonian and C. Fosco—Equal contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58523-5_2) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12363, pp. 18–34, 2020.
https://doi.org/10.1007/978-3-030-58523-5_2

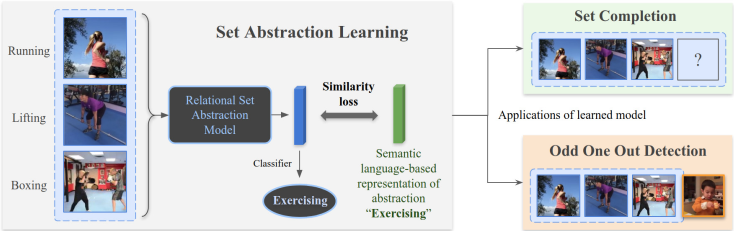
Running
Lifting
Boxing
Set Abstraction Learning
Relational Set Abstraction Model
Similarity loss
Classifier
Exercising
Semantic language-based representation of abstraction "Exercising"
Set Completion
?
Applications of learned model
Odd One Out Detection

Fig. 1. Semantic Relational Set Abstraction and its Applications: We propose a paradigm to learn the commonalities between events in a set (the *set abstraction*) using a relational video model. Our model is trained to approximate the semantic language-based representation of the abstraction and predict the abstract class shared by the videos. Once trained, the model is able to identify the abstraction that represents a set of videos, select videos that fit this abstraction, and detect when a member of the set does not match the common theme.

ability into a computer vision system. Learning the semantic relational abstraction between a set of events (Fig. 1) allows a model to perform cognitive-level tasks similar to a person abstracting common patterns. If a model has learned that *exercising* can take many forms (*running*, *weightlifting*, *boxing*), its feature representation can naturally be used to select which new event is similar to the set, or detect an incompatible exemplar. If the system can enrich the learning of these abstractions with semantic and verbal content, we are one step closer to people's ability to combine visual and contextual relationships to form a deeper understanding of observed events.

In this paper, we propose an approach for learning *semantic relational set abstraction* which recasts a single exemplar recognition task to the task of encoding conceptual relationships shared among a set of videos. We apply our trained model to solve a variety of operations, namely *set abstraction* (what is in common?), *set completion* (which new video goes well with the set?), and *odd one out detection* (which video does not belong to the set?). Additionally, we compare our abstraction model to human performance on a novel *relational event abstraction* task where participants rank a set of query videos according to how closely they align with the abstract semantic relationship found between a set of reference videos. This human baseline provides a strong evaluation metric for determining how well our model can encode semantic relationships and allows us to measure the human ability to abstract common concepts from video sets.

By formulating the new set abstraction task in tandem with a language supervision module, we aim to better approximate human cognitive-level decisions. Importantly, we pioneer a data generation methodology which approximates human behavior in abstraction-related tasks. The semantic relational algorithm allows us to sample training examples that categorize the commonalities between sets of videos in a human understandable way. We root our experiments in event understanding, leveraging the large scale video datasets Kinetics [19] and Multi-

Moments in Time [27], replicating all our results with these two benchmarks. To summarize, the main contributions of this paper are:

1. A novel **relational set abstraction model** which generates representations of *abstract events* relating a set of videos in a language-based geometric space and assigns a human-understandable label to the common concept underlying each set. This can be used for cognitive-level relation tasks, namely *set completion* and *odd one out detection*, achieving human-level performance.
2. A novel paradigm, the **Relational Event Abstraction task**, which measures human performance on event abstraction. Given a set of reference videos representing specific events (e.g. *digging*, *peeling*, *unwrapping*), the task involves finding the common abstract concept shared by the videos in the set (e.g. *removing* in this case), and ranking a set of query videos based on how close they align with the abstraction.
3. A **dataset for Relational Event Abstraction** built using a novel *semantic relational algorithmic methodology* rooted in natural language which correlates highly with human results on the *relational event abstraction* task. This allows us to sample a large number of reference and query sets for training and evaluation, avoiding expensive human annotation.

2 Related Work

Concepts Organization. Concepts can be organized in a hierarchical structure (i.e. trees), a chain (i.e. linear organization), or a ring (i.e. perceptual similarities of colors) [34]. Computer vision work on visual classification most often uses trees, with root categories forming the base of taxonomies (i.e. for object [8] and scene classes [39]). Hierarchies can be pre-defined [17,36] or learned [2,10], and can help with transfer learning between categories [23], and class prediction for videos [29,30]. EventNet [40] built an action concept dataset with a hierarchical structure that includes low-level event labels as leaf nodes and increasingly abstract concept labels as parent nodes. They trained a CNN to identify the low-level event labels from the video frames and combine the representation learned from this model with a set of SVMs to predict the higher-level concepts associated with the video. Here, we similarly use a pre-defined relational organization between activities (i.e. *jog*, *swim* and *weightlift* all share the abstract relation *exercise*) as a tool for learning the abstract semantic relations between sets of videos. While previous works consider a single instance at a time, our goal is to generate representations for *sets* of videos to identify common relationships.

Video Recognition. Two stream convolutional networks (CNNs) [33] combine static images and optical flow. In [11], a recurrent model uses an LSTM to learn temporal relationships between features extracted from each frame. 3D CNNs [35] aim to directly learn motion using 3D convolutional kernels to extract features from a dense sequence of frames. I3D proposes incorporating optical flow with 3D CNNs to form a two stream 3D network [6] "inflated" from 2D filters pre-trained on ImageNet [9]. Temporal Segment [37] and Temporal Relation

Networks [43] model relationships between frames from different time segments while non-local modules [38] capture long-range dependencies. SlowFast Networks [12] combine two streams using dense (fast) and sparse (slow) frame rates to simultaneously learn spatial and temporal information.

Visual Similarity. Prior work has proposed methods for estimating similarity between images and video pairs for retrieval and anomaly detection. Fractal representations have been used to estimate pair-wise image similarity and relationships in order to solve the Odd One Out problem by encoding the spatial transformations needed to convert each image in a set to each other image [24]. Semantic word similarity has been shown to be a good approximation for quantifying visual relationships [41] while Siamese Networks [5] have been used to naturally rank the similarity of sets of images for one-shot image recognition [20]. ViSiL [21] proposes an architecture for learning the similarity between pairs of videos that incorporates both spatial and temporal similarity for video retrieval. Odd-one-out networks [13] learn temporal representations of videos that are used to identify the video in a set that has *out of order* frames. IECO [42] uses ideas from SlowFast Networks [12] to form a two-stream ECO network [44] to learn instance, pose and action representations to estimate video similarity in retrieval.

Learning Semantic Relationships in Visual Data. Different approaches have been proposed for learning semantic relationships between sets of images and videos. Reinforcement learning is used as a method for selecting subsets of data that preserve abstract relationships [28]. A Bayesian model has been used with a conceptual hierarchy formed from ImageNet [9] to identify the concept relation in image sets [18]. Relation Networks have been used to infer object relations [32] and Interaction Networks have helped to identify physical relations in complex systems [3]. We extend the idea of relation learning to videos for learning event relations in sets of varying length rather than object relations between images pairs Laso [1] utilizes similarities, and differences, in object labels of images pairs to improve few shot learning by learning the union, intersection and subtraction between the binary label vectors for each image.

We take a similar approach to learning the intersection operation of the "abstract" labels for sets of videos but we differ in that we operate on sets of varying size and are learning the common semantic abstractions of events shared in the set rather than the common labels found in video pairs. The most similar prior work ranks a set of unseen videos based on their similarity to a provided event description [7] by measuring the semantic correlation between the event and each individual concept in a dictionary of concepts (e.g bike, mountain, etc) with cosine distance on the word embedding generated from a skip-gram model [25] trained on a large text corpus. A set of previously unseen videos is ranked according to the correlation between the video and the provided event using the event-concept correlation and the concept classifier for each video. We utilize word embeddings to capture class relationships as was done in other methods that use similar embeddings to identify semantic visual relations [16,22,41]. However, we introduce a relational graph and abstract relational embeddings that are compounded by the embeddings of related classes (see Sect. 3) and apply our approach to the task of recognizing the abstraction between a set of videos.

3 A Dataset for Relational Event Abstraction

Our goal is to categorize the relationships between sets of videos close to human reasoning. We build a dataset for identifying semantic relational event abstractions between sets using word embeddings from natural language [4] and from an abstraction graph where each node represents a semantic relation between its children. Word embeddings, which capture context and word-to-word relationships [25] from a large text corpus, are complementary to our semantic abstraction graph and allow our model to capture relationships not directly encoded into the graph structure. Next, we provide an overview of our approach.

Video Datasets. As an initial step for representing semantic set relationships, we focus on the *relationships between activities in video clips*, using Kinetics [19] and Multi-Moments in Time (M-MiT) datasets [27]. The labels in Kinetics define specific event classes such as *biking through snow* and *cooking on campfire*. The labels in M-MiT are more general such as *bicycling* and *cooking*. The contrast between the class structures allows us to validate our approach to settings with both low-level event categories (Kinetics) and high-level classes (M-MiT).

Semantic Relational Graph. To form our relational graphs, we start with the activity categories provided by the class vocabularies of Kinetics and M-MiT. We assign each category to a synset in the WordNet lexical database [26] that captures the specific meaning of the class label applied to its member videos. Then we extract the hypernym paths of each class in WordNet and add the path, and the members of each path, to our graph. We verify the graph by hand and add any missing relations not captured by the WordNet paths. Building the graph in this way for the full class vocabulary of each dataset allows us to form a trace from each low-level action to high-level categories that capture the abstract relationships between their descendant classes. For example, in the Kinetics graph, the abstraction node for *baking* has two children that share the relation of *baking*, *making a cake* and *baking cookies*, and is a child of the abstraction node *cooking* together with other categories such as *frying* and *cooking chicken* that share the same relation of *cooking*. We do not restrict the nodes in the graph to have a single parent as we are not building a strict hierarchy but rather a directed relational graph where each node represents abstract semantic relations between its descendants. To illustrate, consider the class *sculpting* from M-MiT which is a child of both *carving*, with *peeling* and *shaving*, and *making art* which includes the descendants *drawing* and *painting* (see Fig. 2). Treating the graph as a hierarchy would be incomplete as we would miss out on the full breadth of relations between these different actions.

Category Embeddings. To increase the amount of information given to our model in training and to solidify the relationship between an abstraction node and its children, we generate a semantic embedding vector based on the intuition of distributional semantic word representations in natural language processing [4,25] for each node. These representations capture contextual word relationships from a large unlabeled corpora. We use the word embeddings generated by the Subword Information Skip Gram (SISG) model [4] which takes into account

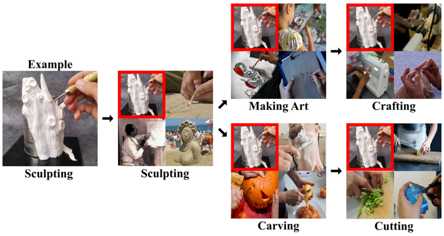

Fig. 2. Semantic Relational Graph: Example of a class relation in our semantic graph for M-MiT. A specific video of *a person sculpting* is an exemplar of the category *sculpting*. The set of videos that share the activity *sculpting* is itself a member of multiple sets that capture higher-level abstractions, like *making art* and *carving* which in turn are members of *crafting* and *cutting* respectively.

the morphology of each word. These vectorized relationships are complementary to our semantic abstraction graph and aid in allowing our model to capture additional relationships not directly encoded into the graph structure.

We begin by assigning each leaf node in our graph the average vector of all the words in the class name using SISG. We then consider this the embedding of that node. From here we traverse the graph and assign each node an embedding vector that is the average of the embeddings of all of its direct children. We use this approach for each parent node to ensure that the embeddings of the abstraction nodes are constrained to capture the common relationships among their children while downplaying features that are specific to a single child node. We describe our approach on using these embeddings to train our model in Sect. 4.

Video Embeddings. As a first step in training video models to capture the abstract semantic relationships between different classes described above we assign embedding vectors to each video according to their class associations. For example, a video in the Kinetics dataset with the class *doing aerobics* will be assigned the vector associated with the graph node *doing aerobics*. This ensures that the models described in Sect. 4 learn representations that align with our relational graph. For a video that contains multiple labels in M-MiT we simply take the average vector for all the classes that belong to the video.

Forming a Training Dataset. In order to train models to accurately capture abstract semantic relationships among a set of videos we must first build a dataset consisting of sets of videos that share common abstractions. To do this we iterate through each parent class in our relational graph and sample four videos that belong to any descendant class of the parent. We use four videos to allow a wide class diversity as using more than four videos in a set commonly results in multiple videos belonging to the same class in both M-MiT and Kinetics. This allows us to balance between having a strong training signal (label diversity) while maximizing computational efficiency. We then generate a label set such that we find the lowest common abstraction shared between the members of each subset in the power set of the set of the videos. The abstraction found for a video set of one is simply the label set for that video in the original dataset. In this way we can train a model to find every abstract semantic relationship present in every combination of the videos in the set greatly increasing our training efficiency over training for different set sizes individually. The labels generated for each subset are then paired with their associated embedding vectors. Due to the flexible modularity of our architecture we are able to reduce the subsets to only contain pairs when efficiency is a concern. For this paper we use the full powerset of the input videos to maximize the learning signal for each training step. For Kinetics we generated one million video sets for training and 50k for validation while for M-MiT we generated five million sets for training and 100k for validation. Training and validation videos were all chosen from the associated training and validation sets of each dataset to preserve the original data splits.

3.1 Human Performance on Event Abstraction

We aim to build a model that identifies relationships similar to how humans recognize abstractions between events. First, we collect a human baseline on a video ranking task where the goal is to rank a set of five *query* videos in order of how closely they align with the abstract relationship between a set of *reference* videos. We compare our trained models to human performance in Sect. 5.2.

Collecting a Human Baseline Dataset. Before we collect human baselines we need to build a dataset for ranking videos according to our relational event abstraction paradigm. We begin similar to the approach used for building our training set and iterate through each abstraction node in our relational graph and select a set of N reference videos (where N can be 1, 2, 3 or 4) that share the abstraction. From here we calculate a shared embedding vector that is the average between the vectors for each reference video and the vector of their shared abstraction node. We then sample five query videos from the dataset sorted according to the cosine distance of their embedding vector and this new reference vector. The goal of this approach is to generate a query set that has at least one video closely aligned with the reference set, one that is very different and three videos that have varying levels of similarity to the reference set. This forms a range of videos with a quantifiable metric based on reference set similarity which we can use to evaluate human and model performance.

Collecting Human Performance. To collect human baseline data, we created the Vidrank game (see Supp.) and used Amazon Mechanical Turk to crowdsource annotations. Players were presented with a “Reference” set of 1–4 videos, and an “Unknown” set (Query) of 5 randomly ranked videos. These videos were labeled: “Least Similar”, “Less Similar”, “Similar”, “More Similar”, and “Most Similar” based on their position. The task was to drag and rearrange the videos in Unknown to a ranking based on each video’s similarity to Reference. To ensure reliable results, we required players to pass “vigilance” rounds, where it was clear that a video should be placed in the “Most Similar” or “Least Similar” position. We collected 40 folds of data for each dataset, 10 questions per fold for a total of 800 human responses[1].

[1] Note that these tasks are challenging for humans, who must disregard similarities across scenes, colors, etc. The model can circumvent this problem as it is trained only on event abstractions.

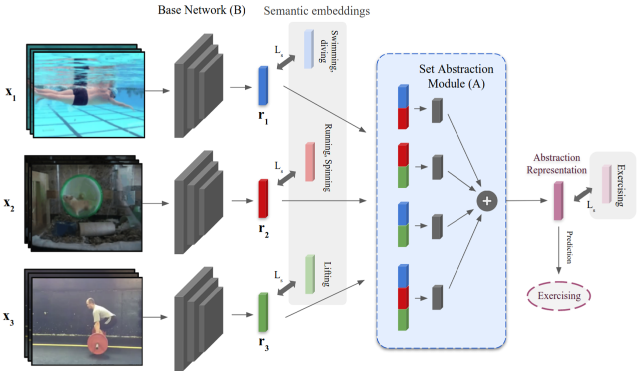

Fig. 3. Set Abstraction Architecture: A set of n videos ($n = 3$ shown) feed into a shared base video model, B. The representations for each video generated by B are combined into each possible subset and fed into the *Set Abstraction Module* (SAM), A, which generates a set-wise representation that is used to identify the common abstractions in each subset. The representations generated by both A and B are contrasted to pretrained semantic embeddings representing the labels for the subset abstractions and the individual videos themselves.

4 Approach

Let x_i be one video, and $\mathcal{X} = \{x_1, \ldots, x_i, \ldots, x_n\}$ be a set of n videos. Given $\mathcal{X}$, our goal is to train a model that correctly predicts the abstract concept that describes all videos in the set and accurately estimates the language model abstraction representation. We will train the model $F(\mathcal{X}; \theta) = (\hat{y}, \hat{e})$ to estimate this category and its semantic word embedding. A naive approach is to first classify each individual video in the set with a traditional video classification network (ignoring learned semantic word embeddings), then use these predictions to look up the lowest common ancestor in the graph. However, this approach is problematic. Firstly, the baseline model will be fragile to errors. If the individual classification model makes a mistake, the set abstraction prediction will be wrong. Secondly, since this does not reason about all elements in the set jointly, this baseline will discard useful information, i.e. the abstract category of a video could change depending on the other videos in the set (which our experiments show). Instead, our approach jointly reasons about all elements in the set.

4.1 Relational Set Abstraction Model

We model $F(\mathcal{X})$ as a deep convolutional network (CNN) that classifies the abstraction of sets of videos. We write F as the composition of two functions A and B: $F(\mathcal{X}) = A\left(\{B(x) | x \in \mathcal{X}\}\right)$ where A is a set abstraction module and B is a base model network for learning individual features.

Video Feature Network (B). This network estimates visual features for each individual video in the set. Several base networks have been proposed to handle temporal and motion information in video [6,11,33,35]. To demonstrate the wide applicability of our framework, we run our experiments with two widely used architectures: a ResNet50 [14] with 3D convolutions (ResNet50-3D) and I3Ds [6]. We observed that ResNet50-3D outperformed I3D in most settings, and thus only report ResNet numbers here for clarity (see Supp. for I3D). Given a set of videos $\mathcal{X} = \{x_0, \ldots, x_n\}$, we use this video feature network to produce a set of features $\mathcal{R} = \{B(x_0), \ldots, B(x_n)\}$, which is fed into the next section. Note that these weights are shared across each video in the set.

Set Abstraction Module (A). Given feature embeddings of each video, our set abstraction module (SAM) is trained to predict the common category of all the videos in the set. Rather than learning a representation for one video, the network learns a representation for a set of videos, in sharp contrast with previous works. To correctly recognize abstract categories, the model must capture the relationships between elements in the set. However, there can be multiple relationship orders: within a single video, across videos, and across higher-order tuples. We model all these relationships by operating on the power set, which is the set of all subsets. Our approach will learn features for each subset and also combine them to produce the final abstraction. Specifically, let $g_k(r_1, \ldots, r_k)$ be a neural network that accepts k inputs, and produces features to represent those inputs. This g network will be able to capture the kth-order relationship between inputs. Given a set of features $\mathcal{R}$, the model's prediction can be written as,

$$A(\mathcal{R}) = h\Big(\sum_{r_i \in \mathcal{R}} g_1(r_i) + \sum_{r_i, r_j \in \mathcal{R}} g_2(r_i, r_j) + \ldots \Big).$$

Each g computes features to capture the relationships between its inputs. We exert two sources of supervision onto this representation: (1) we train a shared linear classifier to predict the common abstraction between the inputs of g and (2) train a separate linear layer to estimate the word embedding of the abstraction. The representations produced by each g_k are summed together to create an order invariant representation and the abstract category for the entire set is estimated by another network $h(\cdot)$. Figure 3 illustrates this module.

4.2 Learning

The video feature network and the set abstraction module can be trained jointly with stochastic gradient descent. As described in Sect. 5.1, to generate the abstract classification of a set of videos we input the representations computed for each video using the video feature network ($\mathcal{R} = \{B(x_0), \ldots, B(x_n)\}$) into our *set abstraction module* (g) and compute the abstract representations for each set of videos (r_i) in the power set of input videos ($\mathcal{R}^2$), $A(\mathcal{R}) = g(r_i) \; \forall \; r_i \in \mathcal{R}^2$.

We then apply these representations to two linear models (h and e) that capture the abstraction class belonging to the set (h) and the category embedding of the abstraction class (e). The embedding provides additional supervision and ensures that the representations generated by the model adhere to semantically and contextually relevant features.

We train the model by averaging the cross entropy losses, $\mathcal{L}_{ce}$, between the predicted abstraction class $(h(g(r_i)))$ and the ground truth class (a_c) and the mean squared error, $\mathcal{L}_{mse}$, between the generated embedding $(e(g(r_i)))$ and the embedding of the ground truth abstraction class (a_e) for each subset (r_i) in the power set of input videos $(\mathcal{R}^2)$:

$$\mathcal{L}_{total} = \frac{1}{|\mathcal{R}^2|} \sum_{r_i \in \mathcal{R}^2} \mathcal{L}_{ce}\big(h(g(r_i)), a_c\big) + \mathcal{L}_{mse}\big(e(g(r_i)), a_e\big).$$

This loss combines the error from generating both the class and the embedding vector of the abstraction class for each possible set of videos given the input set. By doing this we maximize the supervision signal provided from each video input set without recomputing features for different combinations. In practice we train with 4 videos producing 15 different video sets (we omit the empty set).

5 Experiments

We evaluate our *set abstraction model* on three tasks:

1. **Recognizing set abstractions:** Predict the direct relational abstraction for a set of videos.
2. **Set completion:** Given a set of *reference* videos and a set of *query* videos, select the query that best fits in the reference set.
3. **Finding the odd one out:** Identify the video in a set that does not share the relational abstraction common to the other videos in the set.

Experimental Setup. Our hypothesis is that jointly reasoning about all videos in a set will enable models to more accurately predict set abstractions. We compare our *set abstraction model* (3DResNet50+SAM) against a baseline model that maintains the same base model architecture and is trained for standard classification without a set abstraction module (3DResNet50). As a secondary baseline, we train the same base model for multi-label classification using binary cross entropy loss (3DResNet50+BCE) where the labels consist of all the ancestors of the ground truth class, as well as itself, provided by the dataset. We use the standard training, validation and test splits provided by M-MiT and Kinetics and show evaluation results for each task on the corresponding test set.

Comparison to Previous Work. To the best of our knowledge, there are no existing video-based models that explicitly address the set abstraction task. Thus, we compare our model to the following extensions of similar previous work:

- Relation Networks [31]: we replace SAM with their Relation Module that computes representations over pairs of objects, and use the output of f_ϕ as the abstraction representation. Importantly, this module can only work with pairs of videos, so we cannot compute results on set completion with N=1.

Table 1. Recognizing Set Abstractions: Classification accuracy (percent) of the models evaluated on the set abstraction task. Here, N is the number of elements in the set, and the top$k$ chance level is the sum of the frequency of the top$k$ most frequent abstract nodes presented during evaluation.

Dataset	Model	N = 2		N = 3		N = 4	
		Top1	Top5	Top1	Top5	Top1	Top5
M-MiT	Chance	7.9	16.2	7.9	16.2	7.9	16.2
	3DResNet50	17.1	31.2	22.6	38.8	26.0	42.9
	3DResNet50 (BCE)	3.9	30.0	4.9	34.5	5.1	38.0
	3DResNet50+RN [31]	32.4	65.2	39.2	75.4	44.9	82.1
	3DResNet50+SAM (Ours)	**34.0**	**66.9**	**41.1**	**77.1**	**47.2**	**83.8**
Kinetics	Chance	0.44	2.18	0.44	2.18	0.44	2.18
	3DResNet50	29.9	49.1	22.1	42.8	17.9	40.4
	3DResNet50 (BCE)	2.8	25.0	2.2	22.2	0.5	22.5
	3DResNet50+RN [31]	53.9	83.0	61.6	90.2	66.0	93.8
	3DResNet50+SAM (Ours)	**60.5**	**86.0**	**65.3**	**91.6**	**69.9**	**94.6**

- Odd One Out Networks [13]: although their method works on snippets of a single video and only solves the OOO task to generate a representation, we re-purpose their network ($O3N$) to OOO by extending the number of inputs and training to directly predict the input that does not belong to the set.

Implementation Details. We use PyTorch implementations of 3D ResNet50 [15] as the basis for our video feature networks. Each n-scale relation module in SAM A is a two-layer Multilayer Perceptrons (MLP) with ReLU nonlinearities and 2048 units per layer. All models were optimized using stochastic gradient descent with a momentum term of 0.9 and weight decay of 5e-4. An initial learning rate of 0.001 was decreased by a factor of 10 every 20 epochs of training. Models were trained until convergence (∼50–60 epochs).

5.1 Recognizing Set Abstractions

We first evaluate our model on recognizing the abstract category of a set of N videos. We use our abstraction model to directly predict this category given the set. Our baseline model (3DResNet50) individually predicts the specific class category for each video in the set, then computes the set abstraction class directly from the semantic graph (Sect. 3) based on its predictions. Our multi-label baseline (3DResNet50+BCE) also evaluates each video independently and selects the abstract category with the highest mean probability across videos. Our results suggest that there are significant gains by jointly modeling all elements in the video set. Table 1 quantitatively compares the proposed *set abstraction model* against the baselines and Fig. 4 shows qualitative results. Since the margin of improvement increases with the size of the input set, this suggests that set-based training improves the strength of the learning signal by reducing ambiguity.

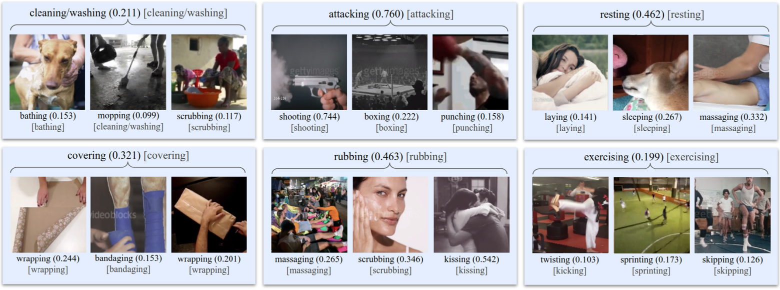
cleaning/washing (0.211) [cleaning/washing]
bathing (0.153) [bathing]
mopping (0.099) [cleaning/washing]
scrubbing (0.117) [scrubbing]
attacking (0.760) [attacking]
shooting (0.744) [shooting]
boxing (0.222) [boxing]
punching (0.158) [punching]
resting (0.462) [resting]
laying (0.141) [laying]
sleeping (0.267) [sleeping]
massaging (0.332) [massaging]
covering (0.321) [covering]
wrapping (0.244) [wrapping]
bandaging (0.153) [bandaging]
wrapping (0.201) [wrapping]
rubbing (0.463) [rubbing]
massaging (0.265) [massaging]
scrubbing (0.346) [scrubbing]
kissing (0.542) [kissing]
exercising (0.199) [exercising]
twisting (0.103) [kicking]
sprinting (0.173) [sprinting]
skipping (0.126) [skipping]

Fig. 4. Qualitative Set Abstraction Results. We show results for sets of length 3, and only show the predictions for the individual videos and the entire set to simplify the visualization. Confidence is indicated in parenthesis and ground truth class in brackets.

Table 2. Set Completion: Rank Correlation of our model (3DResNet50+SAM), a baseline (3DResNet50) and human ranking to the ranking achieved using the embedding distance between a video and the abstraction of a *reference* set of size N on the *set completion* task.

Dataset	Model	N = 1	N = 2	N = 3	N = 4	Avg
	Human Baseline	0.547	0.495	0.595	0.541	0.545
M-MiT	3DResNet50	0.388	0.415	0.455	0.463	0.430
	3DResNet50+RN [31]	–	0.483	0.513	0.533	0.489
	3DResNet50+SAM (Ours)	**0.481**	**0.544**	**0.570**	**0.571**	**0.542**
	Human Baseline	0.432	0.653	0.629	0.606	0.58
Kinetics	3DResNet50	0.339	0.421	0.431	0.459	0.413
	3DResNet50+RN [31]	–	0.491	0.487	0.489	0.489
	3DResNet50+SAM (Ours)	**0.523**	**0.627**	**0.659**	**0.606**	**0.604**

5.2 Set Completion

While recognizing the abstract event relationships shared among a set of videos shows that our model has learned to identify common patterns, we aim to solve more cognitive-level tasks. Thus, we apply our model to the complex task of ranking a set of five *query* videos according to how closely they align to the common abstraction found in a set of $N \in [1, 4]$ *reference* videos (see Fig. 5).

First we use the model to generate an abstract representation of the *reference* set of videos using the method from Sect. 4.1. Then we rank each *query* video according to the cosine distance between this abstract representation and their single video feature representations (Sect. 4.1). This distance tells how closely each video is aligned to the *reference* set abstraction found by our model.

To evaluate our results, we correlate our model's ranking order with the ground truth order found by using the natural-language embedding distance (Sect. 3). In this way we compare the performance to both the abstract rela-

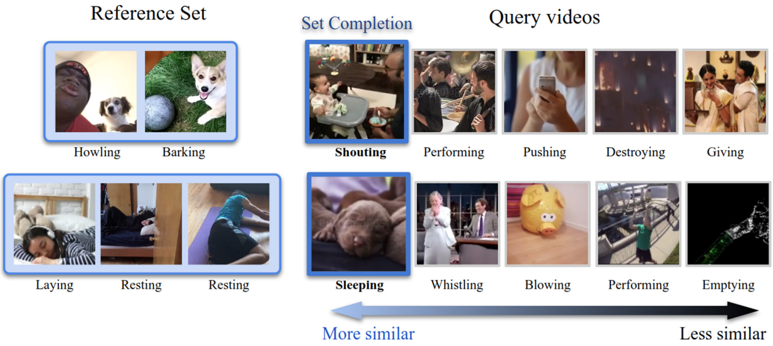

Fig. 5. Set Completion via Learned Abstraction Representations. The videos on the left are the initial reference sets, and the query videos show top ranked retrievals that complete them. The labels of individual videos (not provided to the model) are listed below. The model is able to understand the underlying abstraction regardless of the subject, e.g. choosing the sleeping dog despite all human subjects in the reference.

tionships defined in our *Semantic Relational Graph* and event embeddings associated with the annotated labels of each video. We evaluate the results with, and without, the proposed *abstraction module* (3DResNet50+SAM and 3DResNet50 respectively). For the model without the *abstraction module* we rank the *query* videos using the cosine distance to the average of the feature vectors from each individual *reference* video. Since we are interested in developing models for human-level understanding of abstract relationships, we additionally compare our model results to human performance on the same video ranking task (see Sect. 3). Table 2 summarizes our results. Our abstraction model (3DResNet50+SAM) beats the human baseline on M-MiT when the *reference* set has either 2 or 4 videos and only underperforms the human baseline on Kinetics when the *reference* set has 3 videos. We can see that our model achieves near human performance on M-MIT and surpasses human performance on Kinetics.

5.3 Finding the Odd One Out

Given a set of videos, which video does not belong to the group? With a model trained for set abstraction, we can use its learned internal representation for new tasks, such as identifying the odd one out in a set, without additional training. For example, given a set containing videos of *barking*, *boating* and *flying*, the correct odd one out would be *barking* since *boating* and *flying* are both instances of *traveling*, while *barking* is not.

Given a set of videos, we define a task to identify the odd one out by choosing the video with the largest cosine distance between its video representation and the *abstract* representation of the remaining videos in the set (Sect. 4.1). Intuitively, this method leverages a model's ability to preserve "conceptual" distance in its learned feature space. Table 3 shows that our *set abstraction model*

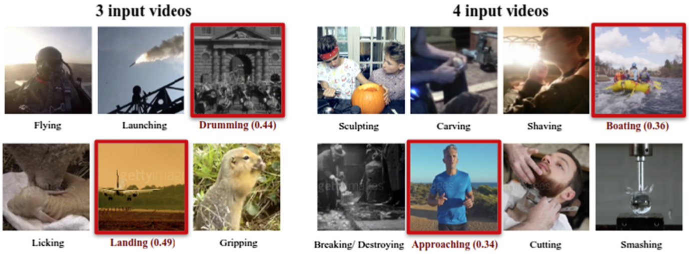

Fig. 6. Qualitative Results for Odd One Out detection: Given sets of three videos (left) and four videos (right), which one is odd? The odd video detected by our *set abstraction model* is indicated by a red bounding box (probability in parenthesis). Even with a small number of other videos to compare to, the model is able to select the odd video out.

Table 3. Odd One Out detection accuracy: Predict the element that does not belong to the set. The language-enhanced features from the set abstraction network are compared with the features from the corresponding base model.

Dataset	Model	N = 3		N = 4	
		Top-1	Top-2	Top-1	Top-2
	Human Baseline	74.21	–	78.04	–
M-MiT	3DResNet50	49.95	78.02	43.73	68.34
	3DResNet50+RN [31]	36.20	64.30	28.94	50.71
	3DResNet50+O3N [13]	34.10	60.11	35.84	60.71
	3DResNet50+SAM (Ours)	**52.63**	**79.86**	**47.21**	**71.11**
	Human Baseline	87.31	–	85.40	–
Kinetics	3DResNet50	65.15	82.65	69.62	81.48
	3DResNet50+RN [31]	40.11	70.97	30.48	54.41
	3DResNet50+O3N [13]	55.14	80.59	66.00	81.80
	3DResNet50+SAM (Ours)	**85.90**	**92.80**	**83.18**	**91.44**

achieves good performance without additional training by making pairwise distance comparisons on subsets of the input. The abstraction model consistently outperformed the baseline model suggesting that our proposed approach indeed learns stronger set representations than the base model. We show some qualitative examples of our model performance in Fig. 6.

6 Conclusion

A central challenge in computer vision is to learn abstractions of dynamic events. By training models to capture semantic relationships across diverse events and predict common patterns, we show that we can learn rich representations of similarity for a new set of tasks. By rooting our models in language, the model can learn abstractions that are better suited to represent how people form high-level classes. Recognizing abstractions should enable vision systems to summarize high-level patterns for different types of applications. While our focus is on capturing action relationships, future work could take into account abstractions involving scenes, objects and other concepts to provide a larger range of relationships to understand events (e.g. "driving" and "jogging" may both occur on a "road" while "writing" and "drawing" may both use a "pencil").

Acknowledgments. This work was supported by the MIT-IBM Watson AI Lab (to R.F and A.O) and NSF IIS 1850069 (to C.V).

References

1. Alfassy, A., et al.: Laso: label-set operations networks for multi-label few-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
2. Bannour, H., Hudelot, C.: Hierarchical image annotation using semantic hierarchies. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2431–2434. ACM (2012)
3. Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., Kavukcuoglu, K.: Interaction networks for learning about objects, relations and physics. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 4502–4510. Curran Associates, Inc. (2016)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://doi.org/10.1162/tacl_a_00051
5. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "Siamese" time delay neural network. In: Proceedings of the 6th International Conference on Neural Information Processing Systems, NIPS 1993, pp. 737–744. Morgan Kaufmann Publishers Inc., San Francisco (1993)
6. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the ICCV (2017)
7. Chang, X., Yang, Y., Hauptmann, A.G., Xing, E.P., Yu, Y.L.: Semantic concept discovery for large-scale zero-shot event detection. In: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI 2015, pp. 2234–2240. AAAI Press (2015)
8. Deng, J., et al.: Large-scale object classification using label relation graphs. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 48–64. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_4
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
10. Deng, J., Satheesh, S., Berg, A.C., Li, F.: Fast and balanced: efficient label tree learning for large scale object recognition. In: Advances in Neural Information Processing Systems, pp. 567–575 (2011)
11. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)

12. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
13. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Jain, M., van Gemert, J.C., Mensink, T., Snoek, C.G.M.: Objects2action: classifying and localizing actions without any video example. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
17. Jia, Y., Abbott, J.T., Austerweil, J.L., Griffiths, T., Darrell, T.: Visual concept learning: combining machine vision and Bayesian generalization on concept hierarchies. In: Advances in Neural Information Processing Systems, pp. 1842–1850 (2013)
18. Jia, Y., Abbott, J.T., Austerweil, J.L., Griffiths, T., Darrell, T.: Visual concept learning: combining machine vision and Bayesian generalization on concept hierarchies. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 1842–1850. Curran Associates, Inc. (2013)
19. Kay, W., et al.: The kinetics human action video dataset. CoRR abs/1705.06950 (2017)
20. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop (2015)
21. Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, I.: Visil: fine-grained spatio-temporal video similarity learning. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
22. Lee, H., Seol, J., Lee, S.: Style2vec: representation learning for fashion items from style sets. CoRR abs/1708.04014 (2017)
23. Lim, J.J., Salakhutdinov, R.R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: Advances in Neural Information Processing Systems, pp. 118–126 (2011)
24. McGreggor, K., Goel, A.: Finding the odd one out: a fractal analogical approach. In: Proceedings of the 8th ACM Conference on Creativity and Cognition, C&C 2011, pp. 289–298. ACM, New York (2011). https://doi.org/10.1145/2069618.2069666
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
26. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to wordnet: an on-line lexical database. Int. J. Lexicography **3**(4), 235–244 (1990)
27. Monfort, M., et al.: Multi-moments in time: learning and interpreting models for multi-action video understanding (2019)
28. Muhammad, U.R., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Goal-driven sequential data abstraction. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
29. Nauata, N., Hu, H., Zhou, G.T., Deng, Z., Liao, Z., Mori, G.: Structured label inference for visual understanding. arXiv preprint arXiv:1802.06459 (2018)

30. Nauata, N., Smith, J., Mori, G.: Hierarchical label inference for video classification. arXiv preprint arXiv:1706.05028 (2017)
31. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Advances in Neural Information Processing Systems, pp. 4967–4976 (2017)
32. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 4967–4976. Curran Associates, Inc. (2017)
33. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
34. Tenenbaum, J.B., Kemp, C., Griffiths, T.L., Goodman, N.D.: How to grow a mind: statistics, structure and abstraction. Science **31**, 1279–1285 (2011)
35. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
36. Verma, N., Mahajan, D., Sellamanickam, S., Nair, V.: Learning hierarchical similarity metrics. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2280–2287. IEEE (2012)
37. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
38. Wang, Y., Hoai, M.: Pulling actions out of context: explicit separation for effective combination. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
39. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: Sun database: exploring a large collection of scene categories. Int. J. Comput. Vis. **119**(1), 3–22 (2016)
40. Ye, G., Li, Y., Xu, H., Liu, D., Chang, S.F.: EventNet: a large scale structured concept library for complex event detection in video. In: Proceedings of the 23rd ACM International Conference on Multimedia, MM 2015, pp. 471–480. ACM, New York (2015). https://doi.org/10.1145/2733373.2806221
41. Aytar, Y., Shah, M., Luo, J.: Utilizing semantic word similarity measures for video retrieval. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2008. https://doi.org/10.1109/CVPR.2008.4587822
42. Zhao, Z., et al.: Instance-based video search via multi-task retrieval and re-ranking. In: The IEEE International Conference on Computer Vision (ICCV) Workshops, October 2019
43. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 831–846. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_49
44. Zolfaghari, M., Singh, K., Brox, T.: ECO: efficient convolutional network for online video understanding. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 713–730. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_43