Embedding Task Structure for Action Detection

Michael Peven Gregory D. Hager
Johns Hopkins University
{mpeven,hager}@jhu.edu

Abstract

We present a straightforward, flexible method to enhance the accuracy and quality of action detection by expressing temporal and structural relationships of actions in the loss function of a deep network. We describe ways to represent otherwise implicit structure in video data and demonstrate how these structures reflect natural biases that improve network training. Our experiments show that our approach improves both accuracy and edit-distance of action recognition and detection models over a baseline. Our framework leads to improvements over prior work and obtains state-of-the-art results on multiple benchmarks. The code is available here.

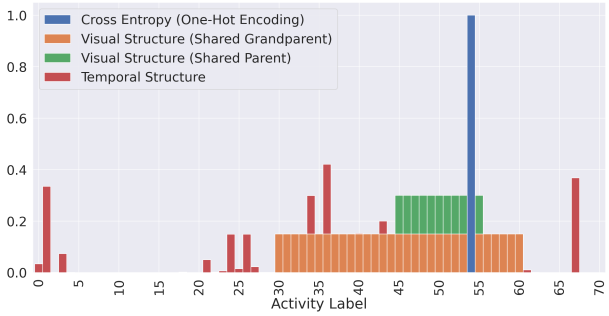
1. Introduction

Which activities are more alike? Peeling a carrot, cutting a carrot, or washing your hands. The answer is obvious, yet cross entropy, the standard loss function used to train a neural network, implicitly represents all three as equally similar or different.

In practice, the consequence of using a one-hot encoding in cross-entropy loss is minimal when enough training samples are seen [32, 48]. Indeed, the ability of deep networks to learn latent relationships automatically from a large enough dataset is the reason they are so ubiquitous. Likewise, complex spatio-temporal patterns in videos can be learned from sufficiently large data sets as evidenced by the fact that the performance of video classification models have become commensurate to their image classification counterparts [71]. Video classifiers have followed the same trend as image classifiers to learn complex patterns: deeper models with hundreds of layers and hundreds of millions of tunable parameters. It is only possible to train these networks because of the large-scale video datasets that have been released in recent years [24, 25, 30].

However, as the temporal span of the inputs increase from image classification ($t = 1$) to activity recognition ($1 < t \lesssim 64$), to action detection[1] ($t \gg 64$), it becomes more difficult for neural-based approaches to abstract high-level relationships directly from data [26]. While more data can possibly address this gap, an alternative solution to this is to embed fundamental *a priori* relationships into the training process. In particular, many visual datasets used for computer vision tasks provide a graphical representation of semantic relationships in the label set. In image classification datasets, relationships are typically represented using a noun-based hierarchy. For example, labels in ImageNet [13] are expressed using the WordNet [47] semantic tree (X is a 'kind of' Y) and the biological labels in [65] are placed in a Linnaean taxonomy. In video datasets like [58], short-

(a) Frames from the activities: peeling carrot, cutting carrot, and washing hands.



(b) Structure embedded into the ground-truth label distribution.

Figure 1. How can we use known relationships to improve action detection models? The images in (a) of cutting and peeling a carrot have a shared object and occur moments after each other in this video. Using the one-hot encoding in (b) does not capture these visual and temporal relationships.

[1] We use the definition of action detection as the classification of all frames in an untrimmed video. This is related to the task of temporal action localization (TAL) which allows for overlapping activity segments.

term temporal relationships are represented in a verb-based hierarchy, and the datasets in [4] and [10] provide the trees for both noun-based and verb-based hierarchies. Long-term structure describing label sequences (e.g. the natural order of making dinner is the sequence: prep work, cooking, plating, clean-up) is given as a state-machine in [1], or can be derived from the labels in the training samples.

In this paper, our goal is to distill *a priori* knowledge about activities into action detection models. We present methods for using a hierarchical label structure (Fig. 2a) and temporal sequence information (Fig. 2b) into a better estimate of the ground-truth label distribution. Previous works have investigated the use of noun-based hierarchies, verb-based hierarchies, and temporal structure. However, they have largely been treated as independent. Our aim is to bridge this gap by creating a unifying framework to embed all forms of structure into an action detection model. The main contribution of this work is demonstrating that a straightforward way to take advantage of task structure can improve performance for action detection, leading to improvements over both our baselines and prior works. For computational efficiency we use light-weight models in our experiments; despite this, our framework obtains state-of-the-art results on multiple evaluation datasets.

1.1. Related Work

Image-based Work Approaches using single-image inputs (image classification, object detection) have investigated using noun-based structural relationships since the release of ImageNet [13], which provided a hierarchical label-set. An intuitive method for encoding prior knowledge of relationships is to embed it in the classification loss function [5, 6, 9, 12, 74]. Similarly, Zhang *et al*. [72] use this approach in a contrastive learning framework. Structure can be used without an explicit hierarchy, Fergus *et al*. [18] enforce consistency with the semantic structure encoded in word embeddings of labels. Likewise, Deselaers & Ferrari [14] take advantage of both semantic and visual distance to train a classifier, and McAuley *et al*. [46] use weak supervision of semantic relationships with latent variables. In contrast to these, Russkovsky & Fei-Fei [57] find that semantic relationships may not actually correspond to useful visual similarities. Rieger *et al*. [56] use prior knowledge to modify a loss function to directly penalize a model's features when they are considered unimportant for the prediction task. Redmon & Farhadi [54] propose an alternative technique of using hierarchical relationships for combining the images from ImageNet, which has a rich hierarchy of objects, and COCO [39], which has a smaller set of general labels for common objects.

Temporal Structure in Video Classification of videos (as opposed to images) allows for pattern recognition in the additional temporal dimension. Early methods for activity recognition [19, 33, 49, 66, 67] use a bag-of-words style approach to model activities as a hierarchy of movement patterns. Later work using deep networks [8, 60, 64] avoid any explicit modeling and leave this kind of short-term structure latent. The release of video datasets involving complicated dynamics and a larger temporal range [24, 25, 30], has pushed this trend even further. Recent transformer based methods [2, 17, 23, 51] are able to model these dynamics but require enormous computational cost. Alternatively, some methods propose to use structure in the label space to balance this trade-off. Bacharidis & Argyros [3] and Leong *et al*. [36] propose method to embed a verb-based tree of activity labels in the hierarchical structure of a neural network. Similarly, Long *et al*. [42] and Surís *et al*. [61] propose the use of hyperbolic embeddings to represent the structure between activities in a continuous domain.

Longer-term Structure Capturing the long-term temporal structure for action detection has been an active field of research long before the modern era of deep neural networks. Early methods of modeling activity sequences [28, 34, 53, 59] largely use some combination of Hidden Markov (or semi-Markov) Models, Conditional Random Fields, context-free action grammars, and Dynamic Bayesian Networks (DBN). More recent work [29, 63] expand upon these methods using features from deep networks to improve performance. Alternatively, the deep networks in [16, 38, 69, 70] learn the long-term structure directly from data. Similar to the method of Camporese *et al*. [7] for action anticipation (future action prediction), we embed sequence structure into the loss function. Our method differs by avoiding both the Markov assumption and marginalization over occurrences in the training data. This allows our framework to model longer-term structure and avoids overfitting to the most frequent sequences. Inspired by the temporal modeling of activities in [45, 55], we use a Poisson point process in order to estimate a distance metric between all activity labels.

Label Distribution Modifications Another set of approaches investigate the label distribution directly. Szegedy *et al*. [62] introduced label smoothing - a modification to the one-hot label distribution that improved performance of their InceptionV3 network. Widespread adoption of this technique has inspired further work [44, 48] toward understanding its effect. Alternatively to parameterizing the label distribution, the approaches in [31] and [52] improved model robustness using label distributions defined by crowd-sourced data (i.e. imperfect labels). Other methods have used label ambiguity for label distribution learning [20, 21, 40, 73], with the objective of obtaining a distributional output for either single-label or multi-label infer-

ence.

2. Methods

Here, we provide a method to combine all forms of structure into a single measure of 'distance' between any two activity labels. Inspired by the work in [5], we use the loss function as a method to embed structure. The loss function is meant to describe the difference between the ground truth and predicted labels. Thus, the distance value we derive can be used with a loss calculation in a straight-forward manner.

2.1. Cross-Entropy Loss

The primary way to train action detection networks is the same as classification models - using cross-entropy loss to calculate gradients for backpropagation:

$$L_{CE}(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{1}$$

Where n is the number of classes, y_i is the ground truth probability for label i, and $\hat{y}_i$ is the model output, the predicted probability of label i.

The standard way to represent the distribution y is through a one-hot encoding: $y_i = 1$ if i is the correct class and $y_i = 0$ for all else. Thus, for this correct class C, Eq. (1) can be simplified as:

$$L_{CE}(y, \hat{y}) = -\log(\hat{y}_{i=C}) \tag{2}$$

One-hot encoding is the result of a deterministic function saying that there is one label associated with the input. The cumulative effect is to form a joint distribution on labels and inputs. However, we can see in Eq. (2) that using standard cross-entropy will only update model parameters based on the values of the correct class only. In other words, the predicted probabilities of all other classes do not matter.

2.2. Tree-based Structure

Using a tree-based representation of activities, we can use the hierarchical structure to create ground-truth distributions that embed this structure. Our goal is to reduce the penalty for predictions when the distance from the ground truth label is small.

We can do this by redefining y_i:

$$\forall c \in C, y_i = \frac{1}{1 + \text{dist}_t(y_c, y_i)} \tag{3}$$

Where dist_t can be any normalized distance function between the predicted label and ground truth. For a tree-based structure (or any graph-like representation) we derive the values using the ordinal rank from a shortest path algorithm.

As long as activity i shares a parent with another activity, there is no longer uniform distribution of 0 over all classes when $i \neq C$. This method is a way to embed informative priors into the ground-truth distributions (see Fig. 1b) for the loss function.

The full loss term using Eq. (3) is:

$$L(y, \hat{y}) = -\sum_{i=1}^{n} \frac{1}{1 + \text{dist}_t(y_c, \hat{y}_i)} \log(\hat{y}_i) \tag{4}$$

2.3. Sequence-based Structure

The classical representation of a temporal event sequence is a deterministic finite-state machine or a more general Markov chain. The method above can be extended to embed such temporal structure into the loss, where the distance function in Eq. (4) is the shortest path between nodes. However, we choose instead to derive sequence statistics from the dataset directly in order to demonstrate that the methods presented here can be generalized to approaches that require no manual specification.

Using training sequences from one of our evaluation datasets, examples of sequence order distributions are shown in Fig. 2b. These are a measure of event occurrences and naturally resemble Poisson distributions. Representing sequences of activities as a Poisson point process allows us to estimate parameters directly from data. Using count data of the minimum number of transitions (i.e. intervening activities) from activity i to activity j, we can obtain a distribution of distances between all activities in the training data. Assuming these are generated from a Poisson distribution, we can obtain the maximum likelihood estimator of the Poisson parameter λ:

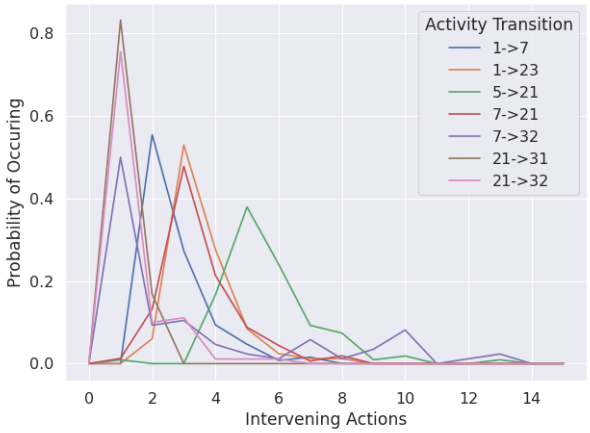
$$\hat{\lambda}_{i \to j} = \frac{1}{N} \sum^{N} x_{i \to j} \tag{5}$$

Where $x_{i \to j}$ is the number of intervening activities between i and j, and N is the total number of observations. We can use this value to obtain a measure of (inverse) distance between all pairs of activities by using the mass function of the Poisson distribution:

$$\text{dist}_s(y_i, y_j) = \begin{cases} 1, & \text{if i=j} \\ \dfrac{e^{-\lambda_{i \to j}} \lambda_{i \to j}^{k}}{k!}, & \text{otherwise} \end{cases} \tag{6}$$

We set $k = 1$ to calculate the (temporal) distance metric as the probability of being a neighboring activity. Using Eq. (6) to determine distance (instead of the prior probabilities) allows us to capture transitions with few examples in the training data because we avoid marginalizing over all transitions. This is useful because we aim to model not just the most frequent sequences, but any valid ordering. For example, if only a small subset of people wash their carrots

(a) Visual structure in the FineGym dataset represented as a tree defined over gymnastics events, event sets, and set elements.



(b) Temporal structure in the IKEA ASM dataset represented as a distribution over the number of intervening activities between two activities.

Figure 2. Representations of activity structure.

before peeling them, those activities should have low distance even if the majority don't. It can now be used as the loss function:

$$L(y, \hat{y}) = -\sum_{i=1}^{n} \text{dist}_s(y_c, \hat{y}_i) \cdot \log(\hat{y}_i) \tag{7}$$

Note that although we use the term 'distance', dist_t is a measure of similarity, and used inversely from Eq. (4), where dist_v is a true distance measure.

Both forms of structure can be summed into a single term:

$$L(y, \hat{y}) = -\sum_{i=1}^{n} \frac{\text{dist}_s(y_c, \hat{y}_i)}{1 + \text{dist}_t(y_c, \hat{y}_i)} \cdot \log(\hat{y}_i) \tag{8}$$

3. Experiments

Distilling activity structure into a taxonomy is up to interpretation. When using a tree-based structure for nouns and verbs, the choice of 'splitting function' defines how the child nodes are placed underneath their parents. For temporal structure, sequence distributions are application dependent - activity ordering could be loosely defined or clear-cut (e.g. furniture building). In order to evaluate how this method holds up to application-based variance, we evaluate results on four distinct action detection datasets that exhibit unique forms of structure.

In these experiments, structure is embedded into models for both activity recognition (trimmed video of a single activity) and action detection (untrimmed video with many possible activities). Details on the architecture, datasets, and implementation are described below.

3.1. Network Architecture

For activity recognition we use a custom implementation of the Temporal Shift Module (TSM) introduced in [37]. The main innovation of TSM is to integrate temporal context by performing channel shifts along the time dimension of the feature map at multiple locations in a CNN. This module is used on top of the framework introduced by Wang *et al*. [68] to allow temporal modeling while maintaining the complexity of a 2D (image-based) CNN. Despite the much lower computational costs, it outperforms many 3D CNNs and is the highest performing baseline in one of our evaluation datasets [58].

For action detection we use a two-stage architecture to classify all frames untrimmed video. We first pretrain the TSM to predict the activity label at the center frame of short snippets sampled from the whole video. Next, this model is applied in a sliding-window manner across all videos, and the features from the final layer of the network are concatenated along the time axis. We use an LSTM [27] for sequence modeling. The video representations are used as input to infer activity labels at each frame in the video.

This choice of architecture fits naturally with our structure distillation process: the visual loss from a tree-based structure in Sec. 2.2 can be used to train the backbone network for activity recognition, and the temporal loss in Sec. 2.3 can be used to train the sequence model. We selected these networks based on computational efficiency of training. Our goal is to evaluate multiple forms of structure in each dataset; the training time of large-scale models is unreasonable with the total combinations of loss functions, backbones, and sequence models. However, our proposed framework is independent of the model at each stage and

is general enough to be applied to any choice of backbone, sequence, or single-stage model for activity recognition or detection.

3.2. Datasets

We select four distinct datasets for experimental evaluation, all of which provide a hierarchical representation of the activity classes.
Activity recognition datasets:

- FineGym [58] contains 4883 videos of different gymnastics events where a three-layer hierarchy is defined (a subset is visualized in Fig. 2a). We are able to use this multi-layer hierarchy to evaluate activity classification when the distance function is defined over activities with a shared parent and activities with a shared grandparent.

- Berkeley MHAD [50] contains 660 videos of actors performing a specified set of actions. The activity are in a three-layer hierarchy defined over motion. For example, *jumping in place* and *jumping jacks* share a parent of *jumping* and a grandparent of *movement in both upper and lower extremities*.

Action detection datasets:

- IKEA ASM [4] contains 371 videos of a furniture assembly task. The structure is given through objects and verbs, given as part of the annotations. These videos exhibit little variation in sequence dynamics because of the procedural nature to putting together furniture.

- EPIC-Kitchens-100 [10] contains over 700 egocentric videos of daily kitchen activities. Visual structure is given using two three-layer hierarchies of {activity, verb, verb-category} and {activity, noun, noun-category}.

3.3. Implementation Details

Both the TSM and LSTM are trained with the AdamW [43] optimizer and a learning rate initialized to $5\mathrm{e}{-4}$. Learning-rate decay (by $\frac{1}{10}$) and early stopping are implemented when validation loss doesn't improve for 5 and 10 epochs respectively. The TSM hyperparameters were set to the same values as [37] and the LSTM has a hidden dimension of 256 units. The batch size was set to 12 videos to keep GPU memory under 12GB. The detection experiments were run for 5 trials to obtain the standard deviations shown in Tab. 2 and Fig. 3. The code is written in Pytorch and all training was performed using a computer with an Intel i9-9900X CPU (3.50GHz), 64GB of DDR4 RAM, and two Nvidia Titan-RTX GPUs.

Model	Accuracy (%)
HFP-I3D [3]	82.89
K-SVM [50]	91.97
H-I3D [3]	96.38
Ours (no hierarchy)	94.55
Ours (Verb 2-level)	96.36
Ours (Verb 3-level)	**98.91**

(a) Results on MHAD.

Model	Gym-99	Gym-288
I3D [58]	74.8	66.7
TSM [58]	80.4	73.5
Hyperbolic [61]	82.54	-
Joint [36]	91.80†	-
Ours (no hierarchy)	88.16	81.89
Ours (Verb 2-level)	**89.50**	**83.34**
Ours (Verb 3-level)	89.14	83.29

† The authors in [36] did not have access to the full dataset and do not provide their evaluation split or code.

(b) Results on both label settings in FineGym.

Table 1. Activity recognition results reported in classification accuracy (% of videos correctly classified) over the test set.

4. Results

4.1. Activity Recognition

To evaluate the effectiveness of our method, we first present results on activity recognition. We know from previous works in Sec. 1.1 that exploiting the label hierarchy should improve classification performance. In addition to verifying this assumption, our aim with these experiments is to answer the following questions: How does our generic method compare against the methods in [3, 36] that implement architectural modifications to embed hierarchical label structure? Furthermore, unlike architectural modifications our method can use a tree-based hierarchy of any depth - does embedding a deeper hierarchy improve performance?

We investigate embedding both the child-parent and child-parent-grandparent relationships in MHAD and FineGym. Results are presented in Tab. 1 measured in average accuracy over all videos in the test split specified in each dataset. Because of the inconsistencies in [36] we include it in Tab. 1b, but disregard their result for comparative purposes. Our experiments demonstrate that embedding the visual structure into the sequence model is enough to obtain state-of-the-art results on both datasets.

Interestingly, the accuracy degrades when moving from a 2 level (child-parent) to a 3 level (child-parent-grandparent)

Dataset	Model	Accuracy (%)
IKEA ASM	VAVA [41]	31.92
	I3D [4]	57.57
	Multi-view CDFL [22]	60.3
	P3D [4]	60.40
	OODL Supervised [22]	63.5
	Ours (No Structure)	66.98 ± 0.93
	Ours (Object)	67.36 ± 0.36
	Ours (Verb)	**68.08** ± **0.28**
EPIC Kitchens	TSM [10]	38.27*
	Ours (No Structure)	35.09 ± 0.10
	Ours (Verb)	35.51 ± 0.62
	Ours (Verb 2-level)	35.80 ± 0.45
	Ours (Object)	35.82 ± 0.43
	Ours (Object 2-level)	**36.23** ± **0.37**

Table 2. Frame-wise accuracy of the model when the backbone classifier is trained using the structure given in IKEA-ASM and EPIC-Kitchens. *The TSM from [10] is trained and evaluated using ground-truth boundaries of the activities, an approximate for the upper bound of action detection.

hierarchy on the FineGym dataset (seen in Tab. 1b). We hypothesize this is due to the top level of the hierarchy in FineGym containing only 4 elements (seen in Fig. 2a). This limited set does not provide enough information to improve results, and could potentially degrade performance for a gymnastics dataset such as this, which contains similar grandchild elements across grandparents (e.g. 'Balance-Beam' and 'Floor Exercise' contain related 'Salto' elements).

4.2. Action detection

In these experiments, our aim is to determine if the features learned from the classification backbone can generalize better when trained with the embedded structure. We use the two-stage model described in Sec. 3.1 to perform action detection in untrimmed videos.

In Tab. 2 we evaluate action detection results when training the backbone model (TSM) using the tree-based loss over the two types of visual hierarchies provided in IKEA-ASM and EPIC-Kitchens: verb (spatio-temporal) and object (spatial only). The LSTM is trained using standard cross-entropy loss in each of these experiments.

The evaluation protocol for action detection on EPIC-Kitchens (called activity detection in [11]) is performed over mean average precision (mAP) results because of the overlapping frames in the segment annotations. Our model assumes a one-to-one relationship between activities and frames; despite this, we obtain an average mAP of 9.93

Loss Function	Accuracy Improvement	Edit Distance Improvement
Temporal	+0.2%	+6.6%
Temporal + Object	+0.6%	+10.3%
Temporal + Verb	-0.6%	-2.0%

Table 3. Ablative experiment measuring difference over the baseline in frame-wise accuracy and edit distance when training the detector with temporal loss and temporal+visual loss. These were performed on IKEA-ASM.

using our non-overlapping segment predictions. This is a significant improvement over the baseline of 5.21 in [10].

The best results on IKEA-ASM are by embedding the structure in the verb-based relationships, but on EPIC-Kitchens the best results are from object-based relationships. We hypothesize this is due to imbalance in the ability of a model to discriminate between the classes of a certain category. For example, the objects are very distinct in IKEA-ASM (shelf vs. table) but most verbs are fairly similar (align vs. attach) and vice-versa in EPIC-Kitchens (fork vs. spoon, wash vs. grate). We observe this across all models by looking at accuracy of parent nodes in the tree: on average, the correct object class is predicted more often in IKEA-ASM than the verb class (74.9% to 68.4%) and likewise for the correct verb parent in EPIC-Kitchens (36.8% to 36.0%). Thus, we see a greater benefit from embedding the less discriminative relationship directly into the training process. These differences in per-category accuracy illustrate the data dependence of performance gains when embedding structure representations.

4.3. Temporal Structure

We found that embedding temporal structure into the sequence model led to no significant difference in predictions once the backbone classifier was already trained using visual structure, $\chi^2_1 = 0.451, P = 0.502$ [15]. We hypothesize that this is because the loss function defined in Sec. 2.3 embeds the structure of short-term dependencies, which are already captured in the temporal range of the TSM. The modified loss will have little effect on the LSTM as the features are discriminative enough without it.

To experimentally evaluate the effect of using temporal structure, we isolate the sequence model by reducing the generalizability of the features from the TSM. We use a TSM pretrained on Kinetics [30] to generate features with less discriminative power. Our goal is to rely more on the temporal model (LSTM) when evaluating the effect of the sequence-based loss function. We measure the relative difference of accuracy and edit distance over a baseline model trained using standard cross-entropy loss. Edit distance is

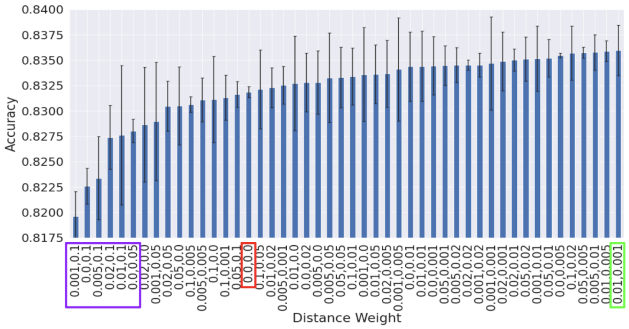


Figure 3. Distance function parameter sweep when using the full hierarchy in the FineGym dataset. The results are sorted by mean accuracy over five trials. The labels are {shared-grandparent, shared-parent} weights used to calculate distance. All weights in the purple box have: shared parent distance > shared grandparent distance. The red box is equivalent to using standard cross-entropy loss with a one-hot encoding. The best performing model is shown in green.

used to measure the 'smoothness' of predictions using a normalized Levenshtein edit score as defined in [35]. Results are in Tab. 3. We observe little change in frame-wise accuracy, but the edit distance of the predictions are much better when using both temporal loss and the combination of temporal and visual (object-based) loss. The reduction in edit distance when using verb-based relationships falls in line with our reasoning above, as motion is the primary discriminator between activities in Kinetics (optical flow inputs outperform RGB [8, 37]).

4.4. Parameter Selection

To evaluate the effect of the parameter selection on the loss function we perform a sweep over the parameter used to calculate the distance metric in Eq. (4). We use the multi-layer tree in FineGym's defined activity structure (see Fig. 2a). Fig. 3 displays accuracy of the model over multiple values of the distance weight for children under a shared parent, and children under a shared grandparent. The results follow our intuition - when the distance weights children under a shared grandparent as 'closer' than children under a shared parent, the model performs worse than using standard cross-entropy loss. When it is the other way around, we see an increase in classification accuracy.

5. Conclusion

We present a simple method to embed relationships between activities into the loss function used to train action detection models. This can be used on the visual structure described through shared verbs or objects, and the temporal structure derived from the sequence of activities. We derive how these structure representations can be embedded into a loss function and demonstrate how this improves both accuracy and edit distance of an action detection model. This small change to the training process leads to state-of-the-art results on challenging benchmarks for action detection.

References

[1] Narges Ahmidi, Lingling Tao, Shahin Sefati, Yixin Gao, Colin Lea, Benjamin Bejar Haro, Luca Zappella, Sanjeev Khudanpur, René Vidal, and Gregory D Hager. A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Transactions on Biomedical Engineering*, 64(9):2025–2041, 2017. 2

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 2

[3] Konstantinos Bacharidis and Antonis Argyros. Extracting action hierarchies from action labels and their use in deep action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 339–346. IEEE, 2021. 2, 5

[4] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 2, 5

[5] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12506–12515, 2020. 2, 3

[6] Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: using hierarchies to improve deep classifiers. *arXiv preprint arXiv:1811.07125*, 2018. 2

[7] Guglielmo Camporese, Pasquale Coscia, Antonino Furnari, Giovanni Maria Farinella, and Lamberto Ballan. Knowledge distillation for action anticipation via label smoothing. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3312–3319. IEEE, 2021. 2

[8] 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. 2, 7

[9] Jingzhou Chen, Peng Wang, Jian Liu, and Yuntao Qian. Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4858–4867, 2022. 2

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection,

pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 2, 5, 6

[11] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, and Michael Wray. Epic-kitchens-100 2021 challenges report, 2021. 6

[12] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, pages 71–84. Springer, 2010. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2

[14] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *CVPR 2011*, pages 1777–1784. IEEE, 2011. 2

[15] Thomas G Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998. 6

[16] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 2

[17] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[18] Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic label sharing for learning with many categories. In *European Conference on Computer Vision*, pages 762–775. Springer, 2010. 2

[19] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *International journal of computer vision*, 107:219–238, 2014. 2

[20] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017. 2

[21] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016. 2

[22] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, Chiho Choi, and Behzad Dariush. Weakly-supervised online action segmentation in multi-view instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13780–13790, 2022.

[23] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16102–16112, 2022. 2

[24] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 1, 2

[25] Chunhui Gu, Chen Sun, David Ross, Carl Martin Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 2

[26] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 1

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[28] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000. 2

[29] Jonathan D Jones, Cathryn Cortesa, Amy Shelton, Barbara Landau, Sanjeev Khudanpur, and Gregory D Hager. Fine-grained activity recognition for assembly videos. *IEEE Robotics and Automation Letters*, 6(2):3728–3735, 2021. 2

[30] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 6

[31] Christoph Koller, Göran Kauermann, and Xiao Xiang Zhu. Going beyond one-hot encoding in classification: Can human uncertainty improve model performance?, 2022. 2

[32] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 1

[33] Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese. Action recognition by hierarchical mid-level action elements. In *Proceedings of the IEEE international conference on computer vision*, pages 4552–4560, 2015. 2

[34] Benjamin Laxton, Jongwoo Lim, and David Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[35] Colin Lea, René Vidal, and Gregory D Hager. Learning convolutional action primitives for fine-grained action recognition. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1642–1649. IEEE, 2016. 7

[36] Mei Chee Leong, Hui Li Tan, Haosong Zhang, Liyuan Li, Feng Lin, and Joo Hwee Lim. Joint learning on the hierarchy representation for fine-grained human action recognition. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1059–1063. IEEE, 2021. 2, 5

[37] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision, pages 7083–7093, 2019. 4, 5, 7

[38] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019. 2

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[40] Miaogen Ling and Xin Geng. Soft video parsing by label distribution learning. *Frontiers of Computer Science*, 13(2):302–317, 2019. 2

[41] Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2181–2191, 2022.

[42] Teng Long, Pascal Mettes, Heng Tao Shen, and Cees GM Snoek. Searching for actions on the hyperbole. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1141–1150, 2020. 2

[43] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[44] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 2

[45] Tahmida Mahmud, Mahmudul Hasan, Anirban Chakraborty, and Amit K Roy-Chowdhury. A poisson process model for activity forecasting. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2016. 2

[46] Julian J McAuley, Arnau Ramisa, and Tibério S Caetano. Optimization of robust loss functions for weakly-labeled image taxonomies. *International journal of computer vision*, 104(3):343–361, 2013. 2

[47] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1

[48] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 1, 2

[49] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 392–405. Springer, 2010. 2

[50] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE workshop on applications of computer vision (WACV)*, pages 53–60. IEEE, 2013. 5

[51] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 2

[52] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9617–9626, 2019. 2

[53] 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

[54] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. 2

[55] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3131–3140, 2016. 2

[56] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International Conference on Machine Learning*, pages 8116–8126. PMLR, 2020. 2

[57] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010. 2

[58] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2020. 1, 4, 5

[59] Yifan Shi, Yan Huang, David Minnen, Aaron Bobick, and Irfan Essa. Propagation networks for recognition of partially ordered sequential action. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004. 2

[60] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 2

[61] Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12617, 2021. 2

[62] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[63] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1250–1257. IEEE, 2012. 2

[64] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE Inter-*

national Conference on Computer Vision (ICCV), December 2015. 2

[65] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1

[66] LiMin Wang, Yu Qiao, and Xiaoou Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 2674–2681, 2013. 2

[67] Limin Wang, Yu Qiao, and Xiaoou Tang. Mofap: A multi-level representation for action recognition. *International Journal of Computer Vision*, 119:254–271, 2016. 2

[68] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 4

[69] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. 2

[70] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 492–510. Springer, 2022. 2

[71] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5):1005, 2019. 1

[72] Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16660–16669, 2022. 2

[73] Zhaoxiang Zhang, Mo Wang, and Xin Geng. Crowd counting in public video surveillance by label distribution learning. *Neurocomputing*, 166:151–163, 2015. 2

[74] Bin Zhao, Fei Li, and Eric Xing. Large-scale category structure aware image categorization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. 2