

# Graph Distillation for Action Detection with Privileged Information

Zelun Luo<sup>1,2\*</sup>Lu Jiang<sup>2</sup>Jun-Ting Hsieh<sup>1</sup>Juan Carlos Niebles<sup>1,2</sup>Li Fei-Fei<sup>1,2</sup><sup>1</sup>Stanford University<sup>2</sup>Google Inc.

## Abstract

In this work, we propose a technique that tackles the video understanding problem under a realistic, demanding condition in which we have limited labeled data and partially observed training modalities. Common methods such as transfer learning do not take advantage of the rich information from extra modalities potentially available in the source domain dataset. On the other hand, previous work on cross-modality learning only focuses on a single domain or task. In this work, we propose a graph-based distillation method that incorporates rich privileged information from a large multi-modal dataset in the source domain, and shows an improved performance in the target domain where data is scarce. Leveraging both a large-scale dataset and its extra modalities, our method learns a better model for temporal action detection and action classification without needing to have access to these modalities during test time. We evaluate our approach on action classification and temporal action detection tasks, and show that our models achieve the state-of-the-art performance on the PKU-MMD and NTU RGB+D datasets.

## 1. Introduction

Recent advancements in deep convolutional neural networks (CNN) have been very successful in various vision tasks such as image recognition [7, 22, 16] and object detection [13, 43, 42]. A notable bottleneck for deep learning, when applied to video, is the lack of massive, clean, and task-specific annotations, as collecting annotations for video is much more time-consuming and expensive. Furthermore, due to various reasons such as privacy concerns or hardware constraints, in many circumstances, only partial modalities (signals) in the video can be observed.

The challenge of lack of data and modalities is encountered in many real-world applications including self-driving cars and health care. A representative example is action detection in surveillance videos, *e.g.* fall detection [39, 68]. First, labeled video clips of fall incidents are difficult to obtain, and secondly, RGB videos, which violate individual privacy, are often unavailable, and hence detection can only be performed on privacy-preserving signals such as depth

\*Work done during an internship at Google Inc.

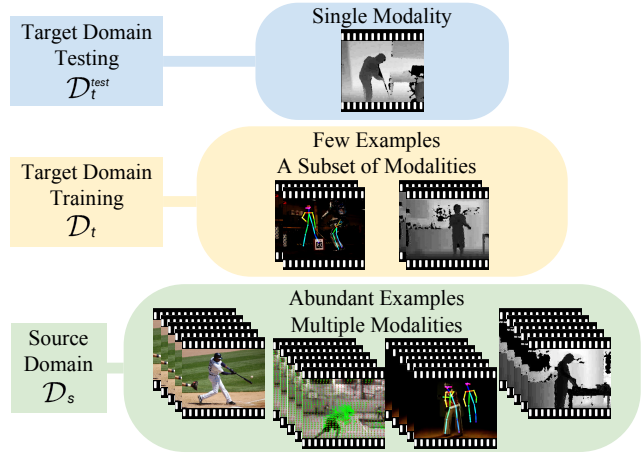


Figure 1: We study the temporal action detection problem under a realistic and challenging condition. In the source domain, we have abundant labeled data from multiple modalities. For the training stage in the target domain, we have very few labeled data and a subset of the source domain modalities. For the testing stage in the target domain, only one modality is used.

videos. This leads to the challenging scenario of training action detectors on a single modality with limited data.

Inspired by this problem, we tackle the video understanding problem in the following setting: we consider temporal action detection under the condition in which we have limited labeled data and partially observed training modalities. To do so, we make use of large action classification datasets that contain multiple *heterogeneous* modalities (our source domain) to assist the training of the action detection model (our target domain), as illustrated in Fig. 1. Our goal is to leverage the rich privileged (auxiliary) information [55, 59, 48] in the source domain and improve model performance on the target domain.

We propose a model that tackles the following technical challenges of our problem: (i) how to effectively utilize multi-modal information in the source domain, and (ii) how to transfer knowledge from source to target. For the first challenge, we propose a *graph distillation* method to distill knowledge from multiple modalities. The distillation is carried out on a graph that is end-to-end learned by optimizing on modality-specific prior and example-specific likelihood. The graph is learned to capture complementary information residing in different modalities, *e.g.*, some actions are easier to recognize by optical flow while others are easier by

Method	Number of modalities		
	src	trg train	trg test
LUPI [48]	-	many	1
Transfer learning [40]	1	1	1
Cross Modal dist. [15]	-	2 (1 unlabeled)	1
Multimodal learning [50]	-	many	same as trg train
Ours	many	subset of src	1

Table 1: Comparison of different learning paradigms. “src” and “trg” stand for source domain and target domain respectively. “dist.” stands for distillation. “-” indicates that the domain is not used. For cross modal distillation, the modality used in test time is unlabeled during training.

skeleton data. For the second challenge, we transfer the rich information acquired from our graph distillation to the target domain. At training time, our target model is able to learn information of additional data and modalities transferred from the source, and at test time leverage this information without needing the auxiliary modalities.

As shown in Table 1, the difference between our method and relevant approaches lies in the fact that we leverage a large source domain dataset with multiple modalities. Our method is reminiscent of the idea of knowledge distillation [17, 33] and related to cross modal distillation [18, 15]. An important contribution of the paper is the novel graph distillation method for both multi-modal action detection and classification. It advances existing methods [17, 33, 18, 15] that focus on pairwise distillation with a fixed distillation direction, *i.e.* using a strong modality to help a weak modality. Our model deals with multiple modalities by dynamically learning distillation directions over a graph.

We extensively validate our method and empirically show that it outperforms the state-of-the-art approaches for action classification and temporal action detection on two public benchmarks. Notably, it improves the state-of-the-art from 84.2% to 90.3% by an absolute 6.1% gain on PKU-MMD [28], and yields an absolute improvement of 4.6% on NTU RGB+D [45]. Our ablation studies reveal that the outstanding performance attributes to using graph distillation to incorporate privileged multi-modal information. To summarize, the contribution of this paper is threefold:

- We study a realistic condition for video understanding with limited labeled data and modalities. We discover that in addition to transferring knowledge from a large source domain dataset, we should also leverage extra multiple modalities.
- We propose a novel graph distillation method for action detection and classification that effectively incorporates multiple heterogeneous modalities of video during training, improving performance at test time when only a single modality is available.
- Our method achieves the state-of-the-art results on two benchmarks, including temporal action detection on PKU-MMD [28] and action classification on NTU RGB+D [45].

## 2. Related work

**Action classification and temporal action detection.** The field of action classification for RGB videos has been studied by the computer vision community for decades. While previous work using hand-crafted features [6, 24, 25, 56] have shown promising results, deep convolutional neural networks have become the dominant approach in recent years [20, 50, 9, 53, 3]. Inspired by the progress in action classification, recent work has tackled the temporal action detection task. Previous work includes frame-level or sliding window classification methods [14, 19, 36], proposal-based methods [11, 49, 69], and end-to-end single-pass frameworks [63, 2].

The success in RGB video understanding has given rise to a series of studies on action recognition with different modalities [52, 66, 21, 58, 60, 10]. Specifically, with the availability of depth sensors and joint tracking algorithms, extensive research has been done on action classification and detection using RGB-D videos [38, 44, 64, 46, 47] or skeleton sequences [26, 30, 31, 45, 67, 32].

Action classification and temporal action detection are the main focus of this paper. Different from previous work, our proposed model focuses on leveraging abundant training examples and privileged modalities on a source dataset. We show that it benefits action detection when the target training dataset is small in size, and only one modality is available at inference time.

**Video understanding under limited data.** Our work is largely motivated by real-world situations where data and modalities are limited. For example, surveillance systems for fall detection [39, 68] often face the challenge that annotated videos of fall incidents are hard to obtain, and more importantly, recording RGB videos is prohibited due to privacy issues. A simple and direct approach is the technique of transfer learning, which has been widely used in many fields [40, 35]. Specifically for video understanding, it is common to transfer models trained for action classification to action detection.

While these methods are proved to be effective, their source and target domains have the same modality. For instance, the input to both domains must be RGB videos. In reality, video data often contains much more explicit information such as optical flow and depth maps. Our method is able to incorporate the rich multi-modal information in the source domain and transfer to the target domain.

**Learning using privileged information.** Vapnik and Vashist [55] introduced a *Student-Teacher* analogy: in real-world human learning, the role of a teacher is crucial to the student’s learning since the teacher can provide explanations, comments, comparisons, metaphors, and so on. They proposed a new learning paradigm called Learning Using Privileged Information (LUPI), where at training time, additional information about the training example is provided

to the learning model. At test time, however, the privileged information is not available, in which case the Student operates without supervision of the Teacher [55].

Several work employed privileged information (PI) on SVM classifiers [55, 23, 54, 59, 12]. Recently, privileged information has been applied to deep learning in various settings such as PI reconstruction [48, 61], information bottleneck [37], and Multi-Instance Multi-Label (MIML) learning with bag-level PI [62]. More related to our work is the combination of distillation and privileged information, which will be discussed next.

**Knowledge distillation.** Hinton *et al.* [17] introduced the idea of knowledge distillation, where knowledge from a large model is distilled to a small model, improving the performance of the small model at test time. This is done by adding a loss function that matches the outputs of the small network to the high-temperature soft outputs of the large network [17]. Lopez-Paz *et al.* [33] later proposed a generalized distillation that combines distillation and privileged information. This approach is adapted by [18] and [15] to perform cross-modality knowledge transfer. Our graph distillation method is different from prior work in that the privileged information contains multiple modalities and the distillation direction is dynamically learned by a graph rather than being predefined by human experts.

### 3. Method

Our goal is to leverage a source data of abundant labeled examples and multiple modalities to assist training models for a target domain with limited labeled data and modalities. We address the challenge by distilling privileged knowledge in the source data. We first formally define the problem: consider an  $L$ -way classification problem in the target domain with a training set  $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}_t|}$ , where  $(x_i, y_i)$  is a feature-label pair,  $x_i \in \mathbb{R}^d$ ,  $y_i \in [1, L]$  is an integer denoting the class label, and  $|\cdot|$  denotes the set cardinality. For action detection, we use the last class  $L$  to denote the “background class”. Since training data on the target domain is limited, we are interested in learning a model by transferring knowledge from a source domain  $\mathcal{D}_s = \{(x_i, \mathcal{S}_i, y_i)\}_{i=1}^{|\mathcal{D}_s|}$ , where  $|\mathcal{D}_s| \gg |\mathcal{D}_t|$ . The novel element  $\mathcal{S}_i = \{x_i^{(1)}, \dots, x_i^{(|\mathcal{S}|)}\}$  is a set of privileged information about the  $i$ -th sample, where the superscript indexes the privileged modality in  $\mathcal{S}_i$ . As an example,  $x_i$  could be the depth image of the  $i$ -th frame in a video and  $x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \in \mathcal{S}_i$  might be RGB, optical flow and skeleton features about the same frame, respectively.

Our model is learned to minimize the empirical risk on unseen test data on the target domain  $\mathcal{D}_t^{test} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}_t^{test}|}$ :

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}_t^{test}|} \sum_{(x_i, y_i) \in \mathcal{D}_t^{test}} \mathbb{E}[\ell(f(x_i), y_i)], \quad (1)$$

where  $\mathcal{F}$  is a class of functions of  $f : \mathbb{R}^d \rightarrow [1, L]$ .  $\ell$  represents the loss function.

In practice, the empirical risk is often optimized on the seen training data under a specific loss. For video classification, a popular choice for  $\ell$  is the softmax cross entropy loss:

$$\ell_c(f(x_i), y_i) = \sum_{j=1}^L \mathbb{1}(y_i = j) \log \sigma(f(x_i)), \quad (2)$$

where  $\mathbb{1}$  is the indicator function which equals 1 when the condition is true and 0 otherwise;  $\sigma$  is the softmax operation  $\sigma(z_k) = \exp z_k / \sum_{j=1}^L \exp z_j$ . For now we mainly discuss the case where privileged information is only available in the source domain. Section 3.4 will discuss how to extend our method to handle more general cases.

#### 3.1. Privileged knowledge distillation

We propose a distillation method to incorporate privileged information in the source domain. Specifically, we follow a two-step training scheme: we first train a model on  $\mathcal{D}_s$ ; then we fix the visual encoder and fine-tune the remaining model for a new task of the target domain  $\mathcal{D}_t$ . To train a model on the source domain, the following loss function is minimized on training data in the source model:

$$\sum_{(x_i, y_i) \in \mathcal{D}_s} \ell_c(y_i, \sigma(f(x_i))) + \ell_m(x_i, \mathcal{S}_i). \quad (3)$$

The loss consists of two parts: the first term is the classification loss in Eq. (2) and the latter is the imitation loss [17]. The imitation loss is often defined as the cross-entropy loss to the *soft logits*. Suppose  $\mathcal{S}_i$  only contains a single modality, and its class prediction function is  $f_{\mathcal{S}}$ , Hinton *et al.* [17] computed the soft logits by  $\sigma(f_{\mathcal{S}}(\mathcal{S}_i)/T)$  at a high temperature  $T$ , where  $\sigma$  is the softmax operator. As shown, soft logits reveal softer class-probability predictions learned by using privileged information, which would be otherwise hidden to the training of the primary feature.

Existing distillation methods [17, 33, 27, 48] may not be directly applicable to our problem due to an important reason: our privileged information consists of *multiple heterogeneous* modalities, such as RGB, optical flow, depth, and skeleton features. Previous studies employed a predefined distillation direction between two modalities, *i.e.* from a strong to a weak modality. In our problem, complementary modalities can assist the training process of each other dynamically, where distillation directions are impossible to predefine. This dynamic and subtle distillation paradigm cannot be fully exploited by existing methods. We relax the assumption in previous work by i) allowing  $\mathcal{S}$  to contain *multiple heterogeneous* modalities and ii) without needing a predefined distillation direction. We have empirically substantiated this hypothesis in our experiments.

The rest of this section focuses on our distillation method for heterogeneous privileged information and is organized as follows. In Section 3.2, we first discuss a special case of

graph distillation on two modalities. In Section 3.3, we generalize our approach to multiple modalities. In Section 3.4, we discuss its extensions to more general problems.

### 3.2. Graph edge distillation

We first consider a special case of graph distillation where only two modalities are involved. Due to their heterogeneous nature, the *soft logits* [17] prove to be insufficient to capture the salient information. To this end, we employ an imitation loss that combines soft logits and feature representations. For notation convenience, we denote  $x_i$  as  $x_i^{(0)}$  and fold it into  $\mathcal{S}_i = \{x_i^{(0)}, \dots, x_i^{(|\mathcal{S}|)}\}$ . The imitation loss in Eq. (3) then becomes  $\ell_m(\mathcal{S}_i)$ . Given two modalities  $a, b \in [0, |\mathcal{S}|]$  ( $a \neq b$ ), we have hand-designed their network architectures  $\Phi = \{\phi_a^j, \phi_b^j \mid \forall j \in [1, \dots, l]\}$  (see Section 4), where each  $\phi_a^j$  is the  $j$ -th layer representation for modality  $a$ , with the last layer  $\phi_a^l$  denoting the logits and  $\phi_a^{l-1}$  denoting the last feature representation.

The proposed imitation loss consists of the loss on logits  $l_{logits}$  and the representation  $l_{rep}$ . The cosine distance is used on the logits as we found the angle of the prediction to be more indicative than the magnitude for heterogeneous modalities. For the loss on representation  $l_{rep}$ , we measure it by the standard cross entropy loss. The total imitation loss  $\ell_m$  from modality  $b$  to  $a$  is computed by the weighted sum of the logits loss and the representation loss. The loss can be encapsulated into a message  $m_{a \leftarrow b}$  passing from  $b$  to  $a$ , calculated from:

$$m_{a \leftarrow b}(x_i) = \ell_m(x_i^{(a)}, x_i^{(b)}) = \lambda_1 l_{logits} + \lambda_2 l_{rep}, \quad (4)$$

where  $\lambda_1$  and  $\lambda_2$  are hyper-parameters. Note that the message is directional,  $m_{a \leftarrow b}(x_i) \neq m_{b \leftarrow a}(x_i)$

### 3.3. Graph distillation

To handle multiple modalities, we introduce a directed graph  $\mathcal{G}$  of  $|\mathcal{S}|$  vertices, named *distillation graph*, where each vertex  $v_k$  represents a modality and an edge  $e_{k \leftarrow j} \in [0, 1]$  is a real number indicating the strength of the connection from  $v_j$  to  $v_k$ . First suppose we are given a fixed graph, the total imitation loss for the modality  $k$  is calculated from:

$$\ell_m(x_i^{(k)}, \mathcal{S}_i) = \sum_{v_j \in \mathcal{N}(v_k)} e_{k \leftarrow j} \cdot m_{k \leftarrow j}(x_i), \quad (5)$$

where  $\mathcal{N}(v_k)$  is the set of vertices pointing to  $v_k$ . As shown, the distillation graph is an important structure which defines not only the direction but also the weights of the messages passing among modalities. It allows multiple modalities to jointly assist training of a weaker modality. Each individual modality is not necessarily stronger than the weak modality.

To exploit dynamic interactions between modalities, we propose to learn the distillation graph along with the original task in an end-to-end manner. Denote the graph by an adjacency matrix  $\mathbf{G}$  where each element  $\mathbf{G}_{jk} = e_{k \leftarrow j}$ .

Given an example  $x_i$ , the graph is learned by:

$$z_i^{(k)}(x_i) = W_{11}\phi_k^{l-1}(x_i^{(k)}) + W_{12}\phi_k^l(x_i^{(k)}), \quad (6)$$

$$\mathbf{G}_{jk}(x_i) = e_{k \leftarrow j} = W_{21}[z_i^{(j)}(x_i) \parallel z_i^{(k)}(x_i)] \quad (7)$$

where  $W_{11}$ ,  $W_{12}$  and  $W_{21}$  are parameters to learn and  $\parallel$  indicates the vector concatenation. Basically,  $W_{21}$  maps a pair of inputs to an edge weight, and the entire graph is learned by repetitively applying Eq. (7) over all pairs of modalities in  $\mathcal{S}$ . Let  $\mathbf{G}_{j:} \in \mathbb{R}^{1 \times |\mathcal{S}|}$  be the vector of its  $j$ -th row. The distillation graph is normalized by:

$$\mathbf{G}_{k:}(x_i^{(k)}) = \sigma([\mathbf{G}_{k1}(x_i), \dots, \mathbf{G}_{k|\mathcal{S}|}(x_i)]/T), \quad (8)$$

where  $T$  is the temperature. The softmax activation is used for two purposes: to ensure every row of  $\mathbf{G}$  sums up to 1 and encourage learning the graph attention by dispersing nonzero weights over a small number of vertices.

The message passing on distillation graph can be conveniently implemented by adding a new layer to the original network. As shown in Fig. 2a and 2b, each vertex represents a modality and the messages are propagated on the learned distillation graph. In the forward pass, we learn a  $\mathbf{G} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$  by Eq. (7) and (8) and compute the message matrix  $\mathbf{M} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$  by Eq. (4) such that  $\mathbf{M}_{jk}(x_i) = m_{k \leftarrow j}(x_i)$ . The imitation loss to all modalities can be collectively calculated by:

$$\ell_m = \mathbf{G}(x_i) \odot \mathbf{M}(x_i) \mathbf{1}^T, \quad (9)$$

where  $\mathbf{1}_{|\mathcal{S}| \times 1}$  is a constant one vector;  $\odot$  is the element-wise product between two matrices;  $\ell_m \in \mathbb{R}^{|\mathcal{S}| \times 1}$  contains imitation loss for every modality in  $\mathcal{S}$ .

In the backward propagation, the imitation loss  $\ell_m$  is incorporated in Eq. (3) to compute the gradient of the total training loss. The gradient will be used to update the graph learning parameters in Eq. (7) and (8). This layer is end-to-end trained with the rest of the network parameters.

For a modality, its performance on the cross-validation set often turns out to be a reasonable estimator to its contribution in distillation. To this end, we add a constant bias vector  $\mathbf{c} \in \mathbb{R}^{|\mathcal{S}| \times 1}$  in Eq. (8), where  $\mathbf{c}_j$  is set w.r.t. the cross-validation performance of the modality  $j$  and  $\sum_{k=1}^{|\mathcal{S}|} \mathbf{c}_k = 1$ . Therefore, Eq. (9) can be rewritten as:

$$\begin{aligned} \ell_m &= (\mathbf{G}(x_i) + \mathbf{1}\mathbf{c}^T) \odot \mathbf{M}(x_i) \mathbf{1}^T \\ &= \mathbf{G}(x_i) \odot \mathbf{M}(x_i) \mathbf{1}^T + (\mathbf{1}\mathbf{c}^T) \odot \mathbf{M}(x_i) \mathbf{1}^T \\ &= \mathbf{G}(x_i) \odot \mathbf{M}(x_i) \mathbf{1}^T + \mathbf{G}_{prior} \odot \mathbf{M}(x_i) \mathbf{1}^T \end{aligned} \quad (10)$$

where  $\mathbf{G}_{prior} = \mathbf{1}\mathbf{c}^T$  is a constant matrix. Interestingly, by adding a bias term in Eq. (8), we decompose the distillation graph into two graphs: a learned example-specific graph  $\mathbf{G}$  and a prior modality-specific graph  $\mathbf{G}_{prior}$  that is independent to specific examples. The messages are propagated on both graphs and the sum of the message is used to compute the total imitation loss. Note adding a constant vector



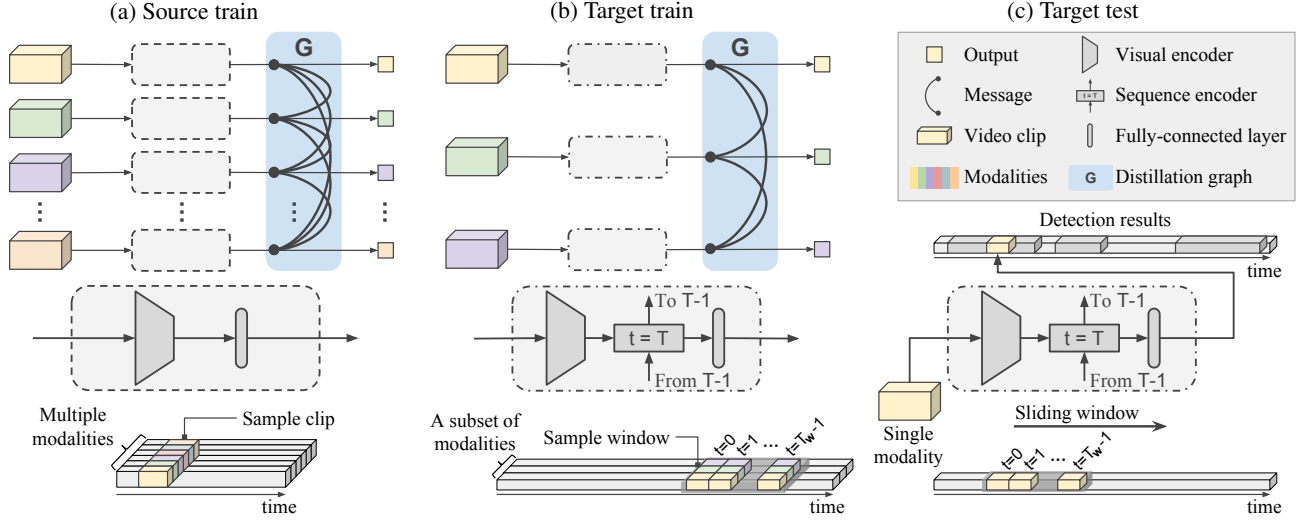


Figure 2: Our proposed network architectures. (a) Action classification with graph distillation in the source domain. The visual encoders for each modality are trained. (b) Temporal action detection with graph distillation in the target domain at training time. In our setting, the target training modalities is a subset of the source modalities (can be 1 or more). Note that the visual encoder trained in the source is transferred and finetuned in the target. (c) Temporal action detection in the target domain at test time, with a single modality.

is more computationally efficient than actually performing message passing on two graphs by Eq. (10). There exists a physical interpretation on the learning process. Our model learns a graph based on the likelihood of observed examples to exploit complementary information in  $\mathcal{S}$ . Meanwhile, it imposes a prior to encourage accurate modalities to provide more contribution. The graph is dynamically learned according to the posterior distribution.

### 3.4. Knowledge transfer and model variants

Regarding transferring a model from the source to the target data, both action classification and detection can be seen as a two-step process: (i) extracting and encoding features; (ii) encoded features are then processed through a classifier to identify the action class or their temporal locations. The first step is shared between two tasks, and is therefore intuitive to use the same architecture to encode the feature (as shown in Fig. 2) for both tasks. In our experiment, the task transfer is done by initializing the visual encoder for action detection with the one learned on the source domain action classification task.

So far, we have only discussed the distillation on the source domain. In practice, our method can also be applied to the target domain on which privileged information is available. Of course, privileged information is never available in test data. This leads to a more general problem where  $\mathcal{D}_t = \{(x_i, \mathcal{T}_i, y_i)\}_{i=1}^{|\mathcal{D}_t|}$ , and, often, the source domain contains richer information, *i.e.*  $\mathcal{T} \subseteq \mathcal{S}$ . When  $\mathcal{T} = \emptyset$ , the problem is identical to the discussed problem. When  $\mathcal{T} \neq \emptyset$ , we transfer the visual encoder of the source domain and then apply the method to minimize Eq. (3) on the target domain training data  $\mathcal{D}_t$ . We found doing so is beneficial even when target training sets are limited in size.

We found jointly train multiple modalities from scratch is nontrivial and can get stuck in bad local minima. To improve the generalization performance, we employ curriculum learning [1] to train the distillation graph. To do so, we fix the distillation graph as an identity matrix in the first 200 epochs. After this stage, we compute the constant  $c$  according to cross-validation and start learning the graph in the end-to-end training. The curriculum learning forces a modality to first focus on utilizing information from its own domain and empirically leads to better performances.

## 4. Action detection and classification models

We evaluate our method on two video understanding tasks: temporal action detection and action classification. In this section, we discuss our network architectures and the details of graph distillation for both tasks.

### 4.1. Visual encoder

The goal of the visual encoder is to encode a short clip of video into a meaningful low-dimensional feature vector. The visual encoder is the key component for both the temporal action detection model and the action classification model. In our multi-modal setting, our data come from two input spaces: image-based data and vector-based input.

**Image encoding.** Let an image-based video clip be  $X = \{x_t\}_{t=1}^{T_c}$ ,  $x_t \in \mathbb{R}^{H \times W \times C}$ , where  $T_c$  is the number of frames in the clip,  $H$ ,  $W$ ,  $C$  are the height, width, and number of channels of the frame, respectively. Similar to the temporal stream in [50], we stack the frames into an  $H \times W \times (N * C)$  input. Note that we do not use the Convolutional 3D (C3D) network [53] because as revealed in [3], it is hard to train especially with limited amount of data. ResNet-18

[16] (with the last fully-connected layer removed) is used to encoder the data.

**Vector encoding.** Let a vector-based video clip be  $X = \{x_t\}_{t=1}^{T_c}$ ,  $x_t \in \mathbb{R}^D$ , where  $T_c$  is the number of frames in the clip, and  $D$  is the vector dimension. Similar to [26], we build a 3-layer GRU framework [5] with time step  $T_c$  to encoder the input. We then compute the feature vector as the average of the outputs of the highest layer across time. The hidden size of the GRU is chosen to be the same as the output dimension of the image-based encoder.

## 4.2. Action detection

Action detection learns to predict whether an action occurs in a video as well as on the temporal extent of when it occurs. Fig. 2b shows the training of our action detection model with graph distillation and Fig. 2c shows the testing during which only a single modality is available. Our architecture for action detection is inspired by [36].

**Training.** We randomly sample 1 window of  $T_w$  video clips from the whole video. Within the window, clips are sampled with length  $T_c$  and step size  $s_c$ . Each clip is fed individually into a visual encoder, followed by a batch normalization layer, a sequence encoder, a batch normalization layer, and a fully connected layer with a softmax activation. Each output is the final class distribution across  $C_d + 1$  classes, where  $C_d$  is the number of action classes and 1 is assigned to the background class. The model is trained to minimize the cross-entropy loss as in Eq. (3) of per-clip classification, and rescaling weight is applied to each class based on its frequency in the training set.

**Testing.** We first uniformly sample windows spanning the whole window with step size  $s_w$ . The prediction of the model is a sequence of class distributions on all the clips (potentially with overlaps depending on the size of  $s_w$ ). This output is post-processed to predict the activity class and temporally localize it. To obtain the temporal localization of the predicted activity class, we first apply a mean filter of  $k$  samples to the predicted sequence to smooth the values through time. Then, the probability of activity (vs no activity) is predicted for each 16-frames clip, being the activity probability the sum of all probabilities of activity classes, and the no activity probability, the one assigned to the background class. Finally, only those clips with an activity probability over a threshold  $\gamma$  are kept and labeled with the previously predicted class. Notice that, for each video, all predicted temporal detections are activity class.

## 4.3. Action classification

The goal of action classification is to classify a trimmed video into one of the predefined categories. Fig. 2a shows the training of our action classification model with graph distillation. Our architecture is inspired by [50].

**Training and testing.** We randomly sample 1 video clip of

Method	Modality	mAP @ tIoU thresholds		
		0.1	0.3	0.5
Deep RGB (DR) [28]	RGB	0.507	0.323	0.147
Deep Optical Flow (DOF) [28]	F	0.626	0.402	0.168
Raw Skeleton (RS) [28]	S	0.479	0.325	0.130
Convolution Skeleton (CS) [28]	S	0.493	0.318	0.121
RS + DR + DOF [28]	RGB + F + S	0.647	0.476	0.199
CS + DR + DOF [28]	RGB + F + S	0.649	0.471	0.199
Qin and Shelton [41]	RGB	0.650	0.510	0.294
Wang and Wang [57]	S	0.842	-	0.743
Ours (graph distillation)	RGB	0.880	0.868	0.801
Ours (graph distillation)	D	0.872	0.860	0.792
Ours (graph distillation)	F	0.826	0.814	0.747
Ours (graph distillation)	S	0.857	0.846	0.784
Ours (graph distillation)	Fusion	<b>0.903</b>	<b>0.895</b>	<b>0.833</b>

Table 2: Comparison of temporal action detection methods on PKU-MMD. Our method achieves the state-of-the-art result. Modality “D”, “F”, “S” stand for depth, optical flow, and skeleton respectively.

Method	Modality	mAP	Method	Modality	mAP
Luo [34]	RGB+D	0.662	Liu [32]	Skel	0.800
Shahroudy [46]	RGB+D	0.749	Ding [8]	Skel	0.823
Liu [29]	RGB+D	0.775	Li [26]	Skel	0.829
Ours	RGB	<b>0.895</b>	Ours	D	0.875

Table 3: Comparison with state-of-the-art on the full NTU RGB+D. For our models, the models are trained on all modalities and tested on a single modality.

length  $T_c$  from a video. The clip is fed into a visual encoder, followed by a batch normalization, and a fully connected layer with a softmax activation. Each output is the final class distribution across  $C_r$  classes, where  $C_r$  is the number of action classes. The model is trained to minimize the cross-entropy loss in Eq. (3) of per-clip classification. No class balancing is applied. For testing, we uniformly sample  $N_r$  clips spanning the entire video. The step size varies depending on the length of the video. The outputs are averaged to obtain the final class distribution across classes.

## 5. Experiments

In this section, we evaluate our method on two benchmarks of rich modalities: action classification on NTU RGB+D [45] and temporal action detection on PKU-MMD [28]. In our experiments, we use NTU RGB+D as our dataset in the source domain, and PKU-MMD as the dataset in the target domain. We demonstrate that graph distillation is effective on both tasks. Furthermore, we show that our method achieves state-of-the-art performance on both datasets.

### 5.1. Dataset and setups

**NTU RGB+D [45]** is one of the largest multi-modal human action classification dataset. It contains 56,880 videos from 60 distinct classes. Each video has exactly one action class and comes with four modalities: RGB videos, depth sequences, 3D joint positions, and infrared frames. In our work, we focus on the cross-subject evaluation protocol introduced in [45], where the training and testing groups

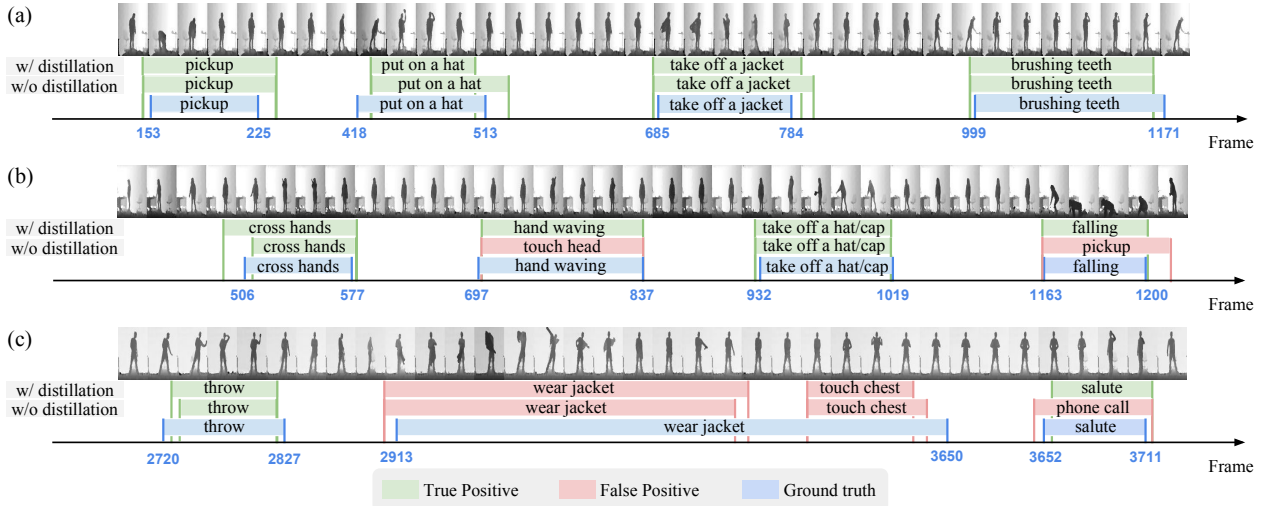


Figure 3: Example predictions generated by our model with and without distillation in the source domain. (a) Both models make correct predictions. (b) The model without distillation in the source makes errors. Our target model learns motion and skeleton information from the source domain, which helps the prediction for classes such as “hand waving” and “falling”. (c) Both models make reasonable errors.

Method	Modality	mAP
No distillation	RGB	0.464
Multi-task [4]	RGB	0.456
Cross-distillation (uniform) [15]	RGB	0.503
Knowledge distillation (uniform) [17]	RGB	0.524
Uniform + cosine	RGB	0.537
Graph + cosine	RGB	<b>0.619</b>

Table 4: Comparison with baseline methods on mini-NTU RGB+D. For our models, the models are trained on all modalities and tested on single modality. “cosine” stands for cosine distance.

are split by action performers. The training and testing sets have 40,320 and 16,560 videos, respectively.

**PKU-MMD** [28] is a large multi-modal action detection dataset. It contains 1,076 long video sequences from 51 action classes. Similar to NTU RGB+D, each sample consists of four modalities: RGB, depth, skeleton, and infrared modalities. Each video contains approximately 20 action instances of various lengths. For evaluation, we compute the temporal Intersection over Union (tIoU) of two time intervals, and evaluate the result by mean Average Precision (mAP) at different tIoU thresholds, the standard metric for action detection [28].

**Modalities.** We include 6 modalities in our experiments: RGB, depth, optical flow, and three skeleton features named Joint-Joint Distances (JJD), Joint-Joint Vector (JJV), and Joint-Line Distances (JLD) [8, 26]. The first three are imaged-based, and the skeleton features are vector-based. RGB and depth frames are provided in the datasets. Optical flow is calculated on the RGB frames using the dual TV-L1 method [65]. For skeleton features, JJD, JJV, and JLD are 3 spatial skeleton features extracted from 3D joint positions, following the method in [8] and [26].

**Baselines.** In addition to comparing with state-of-the-art, we implemented three representative baselines that leverage multi-modal privileged information: multi-task [4], knowledge distillation [17], and cross-modal distillation [15]. For

the multi-task model, we predict the other modalities from the representation of a single modality, and use L2 distance as the multi-task loss. For the distillation method, the imitation loss is calculated as the high-temperature cross-entropy loss on the soft logits in knowledge distillation [17], and L2 loss on both representations and soft logits in cross-modal distillation [15]. The above distillation only supports two modalities, so we assign the uniform weight to average the loss of multiple modalities.

**Implementation details.** For action classification, we train the visual encoder for 200 epochs using SGD with momentum with the learning rate  $10^{-2}$  and decay to 10% at epoch 125 and 175.  $\lambda_1$  and  $\lambda_2$  are set to 10, 5 respectively. For temporal action detection, the visual and sequence encoder are trained for 400 epochs. The visual encoder is trained using SGD with momentum with learning rate  $10^{-3}$ , and the sequence encoder with the Adam optimizer with learning rate  $10^{-3}$ . For both tasks, we down-sample the frame rates of all the datasets by a factor of 3.  $T_c$  and  $T_w$  are both set to 10. For the graph distillation, we apply a softmax with temperature  $T = 0.1$  on the prior. The output dimension of the visual and sequence encoder is set to 512.

## 5.2. Comparison with state-of-the-art

**Temporal action detection.** Table 2 compares our method on PKU-MMD (our target domain) with previous work. The current state-of-the-art method on PKU-MMD uses an RNN on raw skeleton sequence [57]. We train our detection model with pretrained visual encoder from the source domain and with graph distillation. For a fair comparison, we only use skeleton data (JJD, JJV, JLD) at test time. Row 8 and 9 of Table 2 show that our model outperforms previous methods, suggesting that our method can effectively leverage privileged knowledge from a source domain to

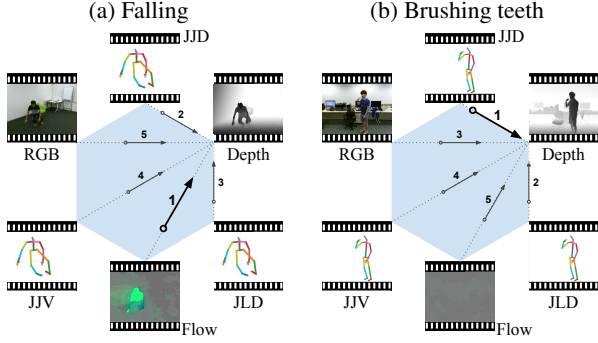


Figure 4: Visualization of distillation graph on NTU RGB+D. The numbers indicate the ranking of the distillation weights, with 1 being the largest and 5 being the smallest. (a) Class “falling”: Our graph assigns more weight to optical flow since optical flow captures the motion information. (b) Class “brushing teeth”: In this case there is almost no motion, and our graph assigns the smallest weight to it. Instead, it assigns the largest weight to skeleton data. For better visualization, only one node is shown.

	mini-NTU RGB+D	mini-PKU-MMD
Graph	mAP / RGB	mAP@0.5 / Depth
Empty graph	0.464	0.501
Uniform graph	0.530	0.513
Prior graph	0.571	0.515
Learned graph	0.619	0.559

Table 5: Comparison of distillation graphs on mini-NTU RGB+D and mini-PKU-MMD. Empty graph trains each modality independently. Full graph uses a uniform weight in distillation. Prior graph is built according to the cross-validation accuracy of each modality. Learned graph is learned by our method.

help train better target models. Other work includes models trained on a combination of RGB, optical flow and skeleton modalities [28]. In particular, row 5 and row 6 of Table 2 are models combining all modalities. Our best model, fusing the predictions of all modalities, yields a remarkable 9% gain on mAP@0.5 over previous work. Fig. 3 illustrates detection results with and without the proposed distillation.

**Action classification.** Table 3 shows the comparison of action classification with state-of-the-art models on NTU RGB+D dataset (our source domain). As we see, our graph distillation model achieves state-of-the-art results on NTU RGB+D with a single modality during test time. The results show that our method, without transfer learning, is still effective for action classification in the source domain.

### 5.3. Ablation studies

In this section, we systematically evaluate the effectiveness of our method across source and target domains. To simulate the scenario where labeled data is limited, we construct mini-NTU RGB+D and mini-PKU-MMD by randomly sub-sampling 5% of training data from the NTU RGB+D and PKU-MMD datasets. For evaluation, we test the model on the entire test set. This section is organized in a series of research questions.

Method		$\theta = 0.1$	$\theta = 0.3$	$\theta = 0.5$
1	trg only	0.248	0.235	0.200
2	src + trg	0.583	0.567	0.501
3	src w/ PIs + trg	0.625	0.610	0.533
4	src + trg w/ PIs	0.626	0.615	0.559
5	src w/ PIs + trg w/ PIs	0.642	0.629	0.562
6	src + trg	0.625	0.610	0.533
7	src + trg w/ 1 PI	0.606	0.592	0.529
8	src + trg w/ 2 PIs	0.637	0.624	0.555
9	src + trg w/ all PIs	0.642	0.629	0.562

Table 6: mAP on mini-PKU-MMD at different tIoU threshold  $\theta$ . The input at test time is depth modality. “src”, “trg”, and “PI” stand for source, target, and privileged information, respectively.

**Is graph distillation effective?** Table 4 shows the comparison to the baseline models detailed in Section 5.1. The results show that our graph distillation method outperforms baseline methods and demonstrate the efficacy of the proposed graph distillation method. By comparing row 2 and 3 in Table 6, we see that the feature trained with privileged information performs better than its counterpart. As discussed in Section 3.4, our distillation can also be applied to the target domain. By comparing row 3 and 5 (or row 2 and 4) of Table 6, we see that performance gain is achieved by applying the same technique in the target domain. The results show that our graph distillation can capture useful information from multiple modalities in both the source and target domain.

**Is learned graph better than other graphs?** In Table 5, we compare the performance of predefined and learned distillation graphs. The results show that the learned graph structure utilizing modality-specific prior and example-specific posterior generates the best results on NTU RGB+D. Fig. 4 shows example distillation graphs learned on NTU RGB+D.

**Is transferring knowledge from the source domain useful?** The comparison between row 1 and row 2 of Table 6 reveals that training the target domain from scratch perform significantly worse than using features pretrained from the source. This shows that transferring knowledge from a larger source domain benefits the target domain.

**Are more modalities useful?** The last 3 rows of Table 6 show that performance gain is achieved by increasing the number of modalities used as the privileged information. It also suggests that modalities offer complementary information during the graph distillation.

### 5.4. Graph distillation on UCF-101

In this section, we consider graph edge distillation - a special case of graph distillation on UCF-101 [51] in which only two modalities (RGB and optical flow) are available. Table 7 shows the action classification results on UCF-101 using the two-stream architecture proposed in [50]. The optical flow modality performs significantly better than RGB when training from scratch. This is consistent with previous findings that dense optical flow is able to achieve very



Method	Modality	mAP	Diff.
From scratch	Flow	0.803	-
From scratch	RGB	0.484	+ 0.000
ImageNet pretrained	RGB	0.728	+ 0.244
Graph Distillation	RGB	<b>0.757</b>	<b>+ 0.273</b>

Table 7: Action classification results on UCF101. For graph distillation model, we distill knowledge from the optical flow stream to the RGB stream.

good performance in spite of limited training data [50]. To testify our method, we train a model on the RGB modality from scratch with distillation. Our distilled model performs much better than the model directly trained from scratch. Note that our distilled model outperforms the fine-tuning model that uses pretrained weights on ImageNet.

## 6. Conclusion

This paper tackled the problem of action detection on limited data and partially observed modalities. We proposed a novel graph distillation method to assist the training on the target domain by leveraging multi-modal privileged information from a large source domain dataset. This was accomplished by our graph-based distillation and task transfer model. The experiments showed that our method can effectively incorporate privileged information from the source domain and train more robust target models. It significantly outperforms several baseline methods and achieves the state-of-the-art for action detection on the PKU-MMD and action classification on NTU RGB+D dataset.

## References

- [1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, pages 41–48, 2009. 5
- [2] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, 2017. 2
- [3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 5
- [4] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998. 7
- [5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. 2014. 6
- [6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1
- [8] Z. Ding, P. Wang, P. O. Ogunbona, and W. Li. Investigation of different skeleton features for cnn-based 3d action recognition. *arXiv preprint arXiv:1705.00835*, 2017. 6, 7
- [9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recur-

- rent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2
- [10] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 2
- [11] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016. 2
- [12] J. Feyereisl, S. Kwak, J. Son, and B. Han. Object localization based on structural svm using privileged information. In *Advances in Neural Information Processing Systems*, pages 208–216, 2014. 3
- [13] R. Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society. 1
- [14] A. Gorban, H. Idrees, Y. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. In *CVPR workshop*, 2015. 2
- [15] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016. 2, 3, 7
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 1, 5
- [17] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS workshop*, 2015. 2, 3, 4, 7
- [18] J. Hoffman, S. Gupta, and T. Darrell. Learning with side information through modality hallucination. In *CVPR*, 2016. 2, 3
- [19] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 2
- [20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2
- [21] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 2
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1
- [23] M. Lapin, M. Hein, and B. Schiele. Learning using privileged information: Svm+ and weighted svm. *Neural Networks*, 53:95–108, 2014. 3
- [24] I. Laptev. On space-time interest points. *IJCV*, 2005. 2
- [25] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2
- [26] C. Li, Q. Zhong, D. Xie, and S. Pu. Skeleton-based action recognition with convolutional neural networks. *arXiv preprint arXiv:1704.07595*, 2017. 2, 6, 7

- [27] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and J. Li. Learning from noisy labels with distillation. In *ICCV*, 2017. 3
- [28] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu. Pku-mmmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017. 2, 6, 7, 8
- [29] J. Liu, N. Akhtar, and A. Mian. Viewpoint invariant action recognition using rgb-d videos. *arXiv preprint arXiv:1709.05087*, 2017. 6
- [30] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016. 2
- [31] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017. 2
- [32] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 2, 6
- [33] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *ICLR*, 2016. 2, 3
- [34] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, 2017. 6
- [35] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei. Label efficient learning of transferable representations across domains and tasks. In *NIPS*, 2017. 2
- [36] A. Montes, A. Salvador, and X. Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint arXiv:1608.08128*, 2016. 2, 6
- [37] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Information bottleneck learning using privileged information for visual recognition. In *CVPR*, 2016. 3
- [38] B. Ni, G. Wang, and P. Moulin. Rgb-d-hudaact: A color-depth video database for human daily activity recognition. In *Consumer Depth Cameras for Computer Vision*, pages 193–208. Springer, 2013. 2
- [39] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007. 1, 2
- [40] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 2
- [41] Z. Qin and C. R. Shelton. Event detection in continuous video: An inference in point process approach. *IEEE Transactions on Image Processing*, 26(12):5680–5691, 2017. 6
- [42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1
- [43] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 91–99. Curran Associates, Inc., 2015. 1
- [44] O. Sener and A. Saxena. rcrf: Recursive belief estimation over crfs in rgb-d activity videos. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. 2
- [45] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016. 2, 6
- [46] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2, 6
- [47] L. Shao, Z. Cai, L. Liu, and K. Lu. Performance evaluation of deep feature learning for rgb-d image/video classification. *Information Sciences*, 385:266–283, 2017. 2
- [48] Z. Shi and T.-K. Kim. Learning and refining of privileged information-based rnns for action recognition from depth sequences. In *CVPR*, 2017. 1, 2, 3
- [49] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2
- [50] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 5, 6, 8
- [51] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, 2012. 8
- [52] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgb-d images. In *AAAI workshop on Pattern, Activity and Intent Recognition*, 2011. 2
- [53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2, 5
- [54] V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of machine learning research*, 16(20232049):55, 2015. 3
- [55] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5):544–557, 2009. 1, 2, 3
- [56] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2
- [57] H. Wang and L. Wang. Learning robust representations using recurrent neural networks for skeleton based action classification and detection. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 591–596. IEEE, 2017. 6, 7
- [58] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012. 2
- [59] Z. Wang and Q. Ji. Classifier learning with hidden information. In *CVPR*, 2015. 1, 3
- [60] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012. 2
- [61] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *CVPR*, 2017. 3

- [62] H. Yang, J. T. Zhou, J. Cai, and Y. S. Ong. Mimi-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information. In *CVPR*, 2017. 3
- [63] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 2
- [64] M. Yu, L. Liu, and L. Shao. Structure-preserving binary representations for rgb-d action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1651–1664, 2016. 2
- [65] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. *Pattern Recognition*, pages 214–223, 2007. 7
- [66] H. Zhang and L. E. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *Intelligent robots and systems (IROS), 2011 IEEE/RSJ international conference on*, pages 2044–2049. IEEE, 2011. 2
- [67] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *WACV*, 2017. 2
- [68] Z. Zhang, C. Conly, and V. Athitsos. A survey on vision-based fall detection. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, page 46. ACM, 2015. 1, 2
- [69] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017. 2