Temporal Query Networks for Fine-grained Video Understanding

Chuhan Zhang
University of Oxford
czhang@robots.ox.ac.uk

Ankush Gupta
DeepMind, London
ankushgupta@google.com

Andrew Zisserman
University of Oxford
az@robots.ox.ac.uk

Abstract

Our objective in this work is fine-grained classification of actions in untrimmed videos, where the actions may be temporally extended or may span only a few frames of the video. We cast this into a query-response mechanism, where each query addresses a particular question, and has its own response label set.

We make the following four contributions: (i) We propose a new model—a Temporal Query Network—which enables the query-response functionality, and a structural understanding of fine-grained actions. It attends to relevant segments for each query with a temporal attention mechanism, and can be trained using only the labels for each query. (ii) We propose a new way—stochastic feature bank update—to train a network on videos of various lengths with the dense sampling required to respond to fine-grained queries. (iii) we compare the TQN to other architectures and text supervision methods, and analyze their pros and cons. Finally, (iv) we evaluate the method extensively on the FineGym and Diving48 benchmarks for fine-grained action classification and surpass the state-of-the-art using only RGB features. Project page: https://www.robots.ox.ac.uk/~vgg/research/tqn/.

1. Introduction

Imagine that you wish to answer particular questions about a video. These questions could be quite general, *e.g.*, "what instrument is being played?", quite specific, *e.g.*, "do people shake hands?", or require a composite answer, *e.g.*, "how many somersaults, if any, are performed in this video, and where?". Answering these questions will in general require attending to the entire video (to ensure that nothing is missed), and the response is *query dependent*. Further, the response may depend on only a very few frames where a subtle action occurs. With such video understanding capability, it is possible to effortlessly carry out regular video metrology such as performance evaluation in sports training, or issuing reports on video logs.

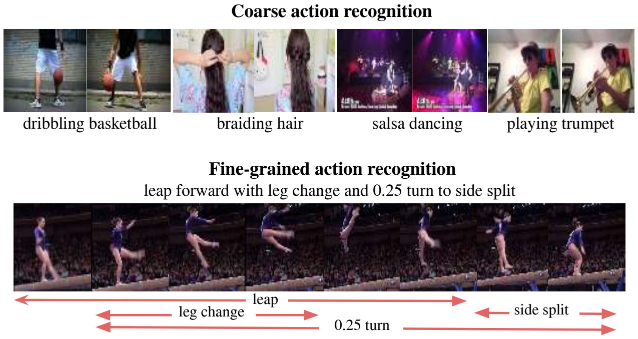
Coarse action recognition
dribbling basketball
braiding hair
salsa dancing
playing trumpet
Fine-grained action recognition
leap forward with leg change and 0.25 turn to side split
leap
leg change
0.25 turn
side split

Figure 1. **Coarse vs. fine-grained action recognition. Top:** Object and background cues from only a few frames can inform classic coarse-grained action recognition in datasets like Kinetics [27], where visually distinct activities are to be distinguished. **Bottom:** However, for finer-grain classification which depends on subtle differences in pose, the specific sequence, duration and number of certain sub-actions, as for the gymnastics sequence above, requires reasoning about events at varying temporal scales and attention to fine details. We develop a novel query-based video network and a training framework for such fine-grained temporal reasoning.

The objective of this paper is a network and training framework that will enable questions of various granularity to be answered on a video. Specifically, we consider *untrimmed videos* and train with *weak supervision*, meaning that at training time we are not provided with the temporal localization information for the response. To this end, we introduce a new Transformer-based [56] video network architecture, the *Temporal Query Network* (TQN), for fine-grained action classification. The TQN ingests a video and a pre-defined set of *queries* and outputs *responses* for each query, where the response is query dependent.

The queries act as 'experts' that are able to pick out from the video the temporal segments required for their response. Since the temporal position of the response is unknown, they must examine the entire duration of the video and be able to ignore irrelevant content, in a similar manner to a 'matched filter' [54]. Furthermore, since the duration of response segments may only be a few frames, excessive temporal aggregation (for example, by average pooling the entire untrimmed video) may lose the signal in the noise.

As the TQN must attend *densely* to the video frames for answering specific queries, and cannot sub-sample in time, we also introduce a *stochastically updated feature bank* so that the model can be trained beyond the constraints imposed by finite GPU memory. For this we use a temporal feature bank in which features from densely sampled contiguous temporal segments are cached over the course of training, and only a random subset of these features is computed online and backpropagated through in each training iteration.

We demonstrate the TQN on two fine-grained action recognition datasets with untrimmed video sequences: FineGym [46] and Diving48 [37]. Both of these datasets share the following challenges: (i) object and backgrounds cannot be used to inform classification, as is possible for more coarse-grained action recognition datasets, *e.g.*, Kinetics [27] and UCF-101 [49] (see Figure 1). (ii) subtle differences in actions, relative spatial orientations and temporal ordering of objects/actors need to be distinguished. (iii) events have a short duration of approx. 0.3 seconds in video clips which are typically 6-10 seconds long, and can be as much as 30 seconds in length. (iv) Finally, the duration and position of events vary and is unknown in training. This lack of alignment between text-description (labels) and videos means that that supervision is weak.

Summary of contributions: (i) we introduce a new model—a Temporal Query Network (TQN)–which enables query-response functionality on untrimmed videos. It can be trained using only the labels for each query. We show how fine-grained video classification can be cast as a query-response task. (ii) We propose a new way—stochastic feature bank update—to train a network on videos of various lengths with the dense sampling required to respond to fine-grained queries. (iii) We compare the TQN to other architectures and text supervision methods, and analyze their pros and cons. Finally, (iv) we evaluate the method extensively on the FineGym [46] and Diving48 [37] benchmarks for fine-grained action classification. We demonstrate the benefits of the TQN and stochastic feature bank update over baselines and with ablations, and the importance of extended and dense temporal context. The TQN with stochastic feature bank update training surpass the state-of-the-art on these two benchmarks using only RGB features.

2. Related Work

Action Recognition. Convolutional neural network have been widely used in action recognition recently, including both 2D networks like the two-stream [48], TSN [59], TRN [74], TSM [40], TPN [69], and 3D networks like LTC [55], I3D [5], S3D [66], SlowFast [13], X3D [12]. Progress in architectures has led to a steadily improved performance on both coarse and fine-grained action datasets [9, 27, 32, 49]. Despite this success, challenges remain: fine-grained action recognition without objects and background biases [8, 37], long-term action understanding [62, 71], and distinguishing actions with subtle differences [46].

Long-Term Video Understanding. Early work used RNNs like LSTM [22] for context-modeling in long videos [36, 38]. More recently, the Transformer [56] architecture has been widely adopted for vision tasks due to its advantage in modeling long-term dependencies. The combination of ConvNets and Transformer is applied not only for images [4, 6, 10, 11, 75], but also on video tasks including representation learning [14, 50, 51], and action classification [17, 60, 62].

Video-Text Representation Learning. Videos are naturally rich in modalities, and text extracted from associated captions, audio, and transcripts is often used for video representation learning. [3, 23] use text as weak supervision to localize actions through alignment, but require text to have the same order as actions. [1, 20] learn to localize and detect action from sparse text labels, while [15] focuses on localizing actions in untrimmed videos by aligning free-form sentences, whereas we learn to answer specific questions with a pre-defined response set. Text is also used in self-supervised text-video representation learning [43, 45, 51], or for supervised tasks like video retrieval [7, 14, 41, 61].

Overcoming Memory Constraints in Frame Sampling. A common way to extract features from a video is by sampling a fixed number of frames, usually less than 64 [5, 40, 66]. However, such coarse sampling of frames is not sufficient, especially for fine-grained actions in untrimmed videos [19, 37, 46, 47]. One common solution is to use pre-trained features[16, 34, 44, 52, 62], but this relies on good initializations and ensuring a small domain gap. While another solution focuses on extracting key frames from untrimmed videos [18, 31].

Visual Question and Answering (VQA). Models for VQA usually have queries which attend to relevant features for predicting the answers [29, 35, 39, 73]. For example, [24] use co-attention between vision and language, and [70] adapts attribute-based attention in LSTM using a pertained attribute detector. [28] proposes a progressive attention memory to progressively prune out irrelevant temporal parts. Our query decoder has a similar query-response mechanism, however, our final goal is action recognition not VQA. Instead of having specific questions for each video, we are interested in a common set of queries shared across the whole dataset.

3. Method

In this section we first, describe the *Temporal Query Network* (TQN) decoder, which given only weak supervision (no event location/duration), learns to respond to the queries

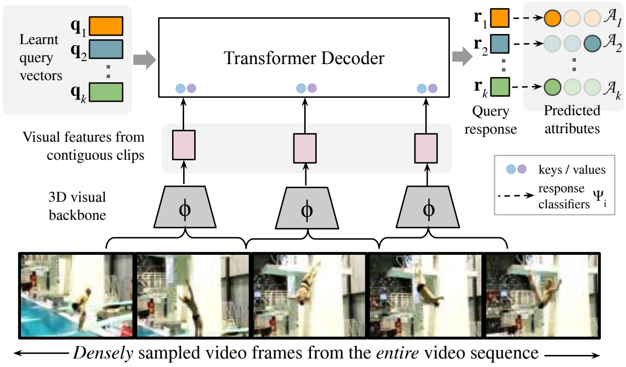

Figure 2. **Temporal Query Network.** A set of permutation-invariant *query vectors* $\mathbf{q}_i$ are learnt for pre-defined queries. They attend over densely extracted visual features in a Transformer [56] decoder and generate *response vectors* $\mathbf{r}_i$, which are linearly classified ($\mathbf{\Psi}_i$) into attributes a_i^j from corresponding attribute sets $\mathcal{A}_i$.

by attending over the entire *densely* sampled untrimmed video (Section 3.1). Second, we introduce a training framework to overcome GPU memory constraints preventing use of temporally-dense video input (Section 3.2). Finally, we explain how the monolithic category labels that are normally provided with fine-grained video datasets, and typically composed of a varying number of sub-labels (or tokens drawn from a finite-set) corresponding to event types and attributes, can be factored into a set of queries and corresponding attributes (Section 3.3).

3.1. Temporal Query Networks

A *Temporal Query Network* (`TQN`) identifies rapidly occurring discriminative events (spanning only a few frames) in untrimmed videos, and can be trained given only weak supervision, *i.e.*, no temporal location or duration information for events. It achieves this by learning a set of permutation-invariant *query vectors* corresponding to pre-defined queries about events and their attributes, which are transformed into *response vectors* using Transformer [56] decoder layers attending to visual features extracted from a 3D ConvNet backbone. Figure 2 gives an overview of the model. The visual backbone and the `TQN` decoder are described below.

Query–Attributes. The query set is $\mathcal{Q} = \{q_i\}_{i=1}^K$, where each query q_i has a corresponding attribute set $\mathcal{A}_i = \{a_1^i, a_2^i, \ldots, a_{n_i-1}^i, \varnothing\}$ consisting of the admissible values a_j^i in response to q_i; $\varnothing$ denotes the `null` value (not present), and the total number of attributes $n_i = |\mathcal{A}_i|$ is query dependent.

For example, in diving videos a query could be the `number of turns` with the attribute set being the possible counts $\{0.5, 1.0, 2.5\}$; or in gymnastics, the query could be the `event type` with attributes `{vault, floor-exercise, balanced beam}`.

Visual backbone. Given an untrimmed video, first visual features for contiguous non-overlapping clips of 8 frames are extracted using a 3D ConvNet: $\mathbf{\Phi} = (\Phi_1, \Phi_2, \ldots, \Phi_t)$, where t is the total number of clips, and $\Phi_i \in \mathbb{R}^d$ is the d-dimensional clip-level visual feature. Note, it is important to extract features *densely* from the *entire* length of the video because: (i) it avoids causing temporal aliasing and also missing rapid events (which span only a few frames), *e.g.*, a somersault, and (ii) selecting a subset of clips from the full video for classification [5, 12, 66] is sub-optimal as the location of these events is unknown.

TQN Decoder. Given the clip-level features and the label queries, the `TQN` decoder outputs a *response* for each query. Concretely, for each label query q_i, a vector $\mathbf{q}_i \in \mathbb{R}^{d_q}$ is learnt for which a response vector $\mathbf{r}_i \in \mathbb{R}^{d_q}$ is generated by attending over the visual features $\mathbf{\Phi}$. Each response vector $\mathbf{r}_i$ is then linearly classified independently into the corresponding attribute set $\mathcal{A}_i$.

In more detail, we use multiple layers of a *parallel non-autoregressive* Transformer decoder, as also used in [4]. Each decoder layer first performs self-attention between the queries, followed by multi-head attention between the updated queries and the visual features. In each attention head, the visual features $\mathbf{\Phi}$ are used to linearly regress *keys* $\Gamma \cdot \mathbf{\Phi}$ and *values* $\Lambda \cdot \mathbf{\Phi}$, where Γ and Λ are the linear key and value heads. The values are gathered using Softmaxed dot-products between the keys and queries as the weights. Finally, a feed-forward network ingests the values from multiple heads and outputs the response vectors. The response vectors from one decoder layer act as queries for the next layer, except for the first layer where the learnt queries $\mathbf{q}$ are input. Hence, each decoder layer refines the previously generated response vectors. Mathematically, if $\ell^{(j)}$ is the jth decoder layer, $j \in \{1, 2, \ldots, M\}$:

$$\begin{aligned}
&\ell^{(j)}(\cdot,\cdot) : \mathbb{R}^{N \times d_q} \times \mathbb{R}^{t \times d} \mapsto \mathbb{R}^{N \times d_q}, \\
&\ell^{(j)}(\mathbf{r}^{(j-1)}, \mathbf{\Phi}) \mapsto \mathbf{r}^{(j)}, \quad \text{and} \\
&\mathbf{r}^{(0)} \triangleq \mathbf{q}.
\end{aligned} \tag{1}$$

The response vectors from the final (Mth) layer $\mathbf{r}_i^{(M)} \in \mathbb{R}^{d_q}$ corresponding to the queries $\mathbf{q}_i$, $i \in \{1, 2, \ldots, K\}$ are classified into the corresponding attribute sets $\mathcal{A}_i$ using K independent linear classifiers $\Psi_i : \mathbb{R}^{d_q} \mapsto \mathbb{R}^{n_i}$, where n_i is the query dependent total number of admissible attributes. Please refer to Figure 2 for a visual representation of this process, and the extended version [72] for details of the Transformer decoder.

Training. The model parameters, *i.e.*, from the visual encoder and the `TQN` decoder are trained jointly end-to-end with the attribute classifiers Ψ_i through backpropagation. The training loss is a multi-task combination of individual classifier losses, which are Softmax cross-entropy $\mathcal{L}_{CE}$ losses on the logits $\Psi_i \cdot \mathbf{r}_i^{(M)}$ over the attribute sets $\mathcal{A}_i$:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{K} \mathcal{L}_{CE}^{(i)}(a^i, \Psi_i \cdot \mathbf{r}_i^{(M)}), \tag{2}$$

where a^i is the ground-truth attribute for the label query q_i.

In essence, the TQN decoder learns to establish *temporal correspondence* between the query vectors and the relevant visual features to generate the response. Since the query vectors are themselves learnt, they are optimized to become 'experts' which can localize the corresponding event in the untrimmed temporal feature stream. Figures 4 and 5 illustrate this temporal correspondence.

Discussion: TQN and DETR. DETR [4] is a recently proposed Transformer based object detection model, which also similarly employs non-autoregressive parallel decoding to output object detections at once. However, there are three crucial differences: (i) the DETR object queries are all equivalent – in that their outputs all specify the same 'label space' (object classes and their RoI), essentially queries are *learnt* position encodings. In contrast the TQN queries are distinct from each other and carry a semantic meaning corresponding to event types and attributes; their output response vectors each specify a different set of attributes, and the number of attributes is query dependent. (ii) This leads to the second difference: since the TQN responses are tied to these queries, they can be trained with direct supervision for attribute labels, thereby avoiding train-time Hungarian Matching [33] between prediction and ground-truth employed in DETR. (iii) Finally, no temporal localization supervision is available to the TQN, while (spatial) locations are provided for DETR training. Hence, although TQN is tasked with (implicit) detection of events, it does so with much weaker supervision.

3.2. Stochastically Updated Feature Bank

Dense temporal sampling of frames for the entire untrimmed video input is key for detecting rapid discriminative events with unknown temporal location. However, this is challenging in practice due to GPU memory constraints which prevent forwarding densely sampled frames in each training iteration. We use a feature memory bank [64, 65] to overcome these constraints.

The memory bank caches the clip-level 3D ConvNet visual features. Note for a given video, the clip features $\mathbf{\Phi} = (\Phi_i, \Phi_2, \ldots, \Phi_t)$, where t is the total number of clips, can be extracted independently of each other. The memory bank is initialized with clip features for all the training videos extracted from a pre-trained 3D ConvNet (details in Section 3.4). Then in each training iteration, a fixed number n_{online} of randomly sampled consecutive clips are forwarded through the visual encoder, *i.e.*, n_{online} clip features are computed *online*. The remaining $(t - n_{\text{online}})$ clip features are retrieved from the memory bank. The two sets of visual features are then combined and input into the TQN decoder for final prediction and backpropagation to update the model parameters. Finally, the clip features in the memory bank corresponding to the ones computed online are replaced with the online features. During inference, all the features are computed online without the memory bank. Figure 3 summarizes the function visually.

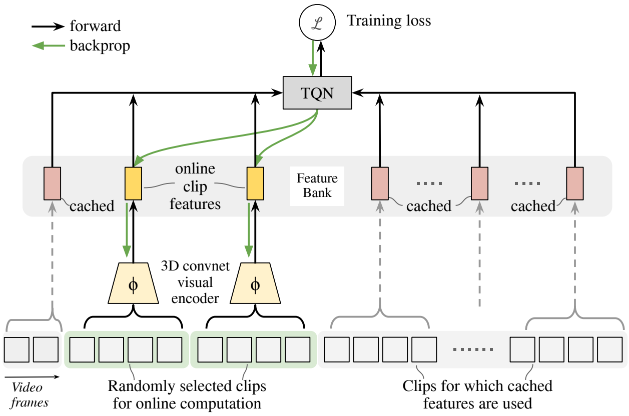

Figure 3. **Stochastically updated feature bank.** Feature banks cache visual encoder features Φ to circumvent GPU memory constraints which prevent forwarding densely sampled video frames from the entire length of the video at each training iteration. Randomly sampled contiguous video clips are forwarded online in each iteration and cached immediately; the rest of the clip features are retrieved from the feature bank. The features are then input into the TQN for *joint* training of both the TQN and the visual encoder. This joint training over dense temporal context is critical for fine-grained performance (Section 5.2).

Advantages. Using a fixed number of clips online decouples the length of videos and the GPU memory budget. As a result our memory bank enables the TQN decoder to be trained (i) jointly with the visual encoder, (ii) with extended temporal context, both of which impart drastic improvements in performance (see Section 5.2). Further, it promotes diversity in each mini-batch as multiple different videos can be included instead of just a single long video.

Discussion: relations with prior memory bank methods. Feature memory banks have been used for compact vector representations of single images [53, 64, 65], whereas we store a varying number of temporal vectors for each video. MoCo [21] and related self-supervised methods [58, 68] update the memory bank features *slowly* *e.g.*, using a secondary network, to prevent representation collapse, whereas given direct supervision for the queries, we can update the features immediately from the single online network. The above works apply memory banks for image/feature retrieval from a large corpus for hard nega-

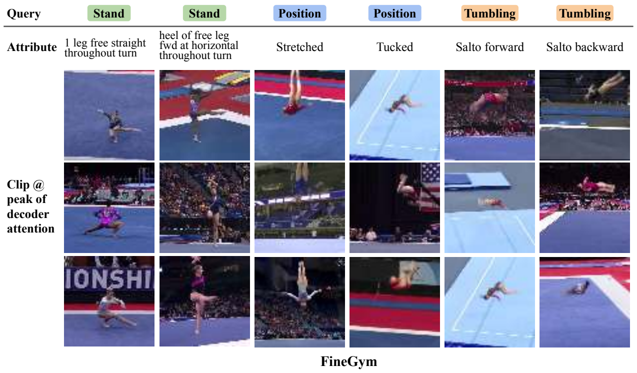
Query
Stand
Stand
Position
Position
Tumbling
Tumbling
Attribute
1 leg free straight throughout turn
heel of free leg fwd at horizontal throughout turn
Stretched
Tucked
Salto forward
Salto backward
Clip @ peak of decoder attention
IONSHI
FineGym

Figure 4. **TQN attention alignment.** For a given query, the `TQN` attends over the clip-level features to generate the responses. We visualize the central frame from the clip with the highest attention score for six different query-attributes from the FineGym [46] dataset. The same queries (but different attributes) are highlighted with a common color. The `TQN` detects and aligns semantically relevant events under variations in appearance and pose without any temporal localization supervision. More visualizations in the extended version [72].

tives in contrastive training, while we use the memory bank for extending the temporal context for each video. Using pre-computed features [34, 44, 52] or *Long-term Feature Banks* [62] are prominent [2, 16, 63] strategies for extending the temporal support of video models. However, all of these works keep the features *frozen*, while we *continuously update* the memory-bank during training. In Section 5.2, we demonstrate these updates are critical for performance.

3.3. Factorizing Categories into Attribute Queries

In this section we illustrate how the pre-defined set of N categories $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ typically associated with fine-grained video recognition datasets can be factored into attribute queries. In such datasets, the categories differ in subtle details *e.g.*, the specific type, duration, or count of a certain sequence of events. These can be rapidly occurring (short duration) events with unknown temporal location and duration (see Figure 1).

The textual descriptions of categories $\mathcal{C}$ are strings composed of a varying number of sub-labels (or tokens drawn from a finite-set) corresponding to event types and attributes (sub-label categories). We leverage this string structure to form queries $\{q_i\}_{i=1}^{K}$ corresponding to sub-label categories, each with an associated attribute set $\mathcal{A}_i$ composed of sub-labels, such that the categories $\mathcal{C}$ can be expressed as a subset of the cartesian product of the attribute sets: $\mathcal{C} \subseteq \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_K$. An example label factorization for four categories is given in Table 1, where four action categories are expressed as a product of two attribute sets containing two and three attributes respectively. Factorization details for the evaluation datasets used in this paper are given in Section 4 and the extended version [72].

Category ↓	**query** →	q_1: leap and jump type		q_2: num turns		
	attribute→	switch leap	split jump	0.5	1.0	∅
switch leap w/ 0.5 turn		✓		✓		
switch leap w/ 1 turn		✓			✓	
split jump w/ 1 turn			✓		✓	
split jump			✓			✓

Table 1. **Illustration of query-attribute factorization of fine-grained action categories.** Four categories are factored into two queries q_1, q_2 with two and three attributes respectively.

This factorization unpacks the monolithic category labels into their semantic constituents (the queries and attributes). It improves data-efficiency, through sharing video data across common sub-labels (instead of disjoint category-specific data), and induces `TQN`-style query-based temporal localization and classification video parsing models.

3.4. Implementation Details

We describe key model and training details below, with further details in the extended version [72].

Model architecture. We use S3D [66] as visual backbone, operating on non-overlapping contiguous video clips of 8 frames each of size 224 × 224 pixels with consistent temporal stride s (s=1 in FineGym, s=2 in Diving48), to output one feature vector per clip. The decoder consists of four standard pre-normalization [67] Transformer decoder layers [56], each with four attention heads, and 1024-dim keys, (learnt-) queries, and values. Dropout rate is 0.1 in the decoder and 0.5 for output features.

Training. The visual encoder is pre-trained on Kinetics-400 [27]. Then, we proceed by a two stage curriculum. First, the model is trained *end-to-end* on *short* videos containing fewer than K frames (FineGym: $K = 48$, Diving48: $K = 128$; as the latter contains approx. 3× longer videos), such that they can fit on two Nvidia RTX 6000 GPUs with batch size 16. Second, the model is trained on the whole training set using the stochastically updated memory bank (Section 3.2) to accommodate long videos. We use the Adam optimizer [30], and train for 50 epochs in the first stage, followed by 30 more epochs in the second.

4. Datasets, Baselines, Label Sets, and Metrics

We evaluate `TQN` for fine-grained action recognition on two video datasets, namely, FineGym [46], and Diving48 [37]. We introduce the datasets, list the baselines methods we compare against, detail the query-attribute based label sets for them, and state the evaluation measure below.

FineGym. FineGym is a recently introduced dataset (2020) for fine-grained action understanding, consisting of HD gymnasium classes with subtle motion details. We evaluate for classification under two settings specified in the

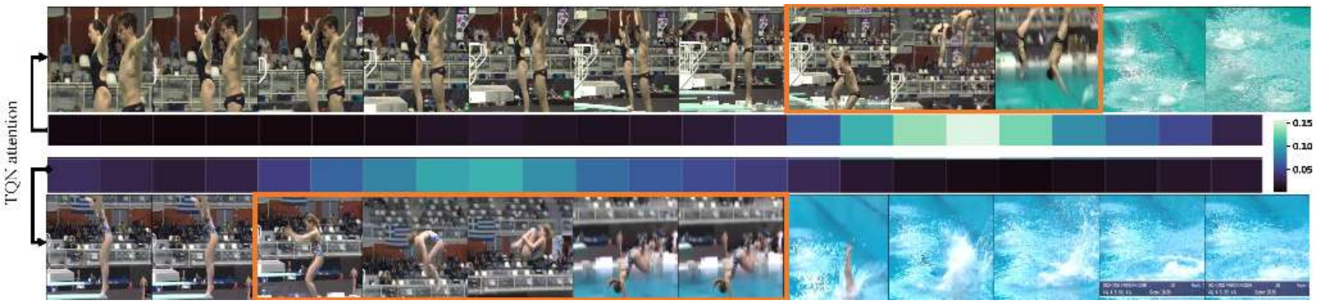

Figure 5. **TQN temporal attention.** **Blue** colored maps visualize the attention averaged over all queries predicted by TQN for two clips from the Diving48 dataset [37]. The peaks in these maps correspond to temporal location of diving 'flight' highlighted in **orange**. TQN rejects non-informative frames at the start and end of untrimmed videos to localize discriminative frames relevant for fine-grained recognition.

dataset with standard train/eval splits: (1) Gym99: relatively balanced data for 99 categories with 26k training/8.5k testing videos; and (2) Gym288: long-tailed data for 288 categories with 29k training/9.6k testing videos. There is a large variation in video lengths: min: 13 frames, max: 877 frames, average: 47 frames.

Diving48. Diving48 contains competitive diving video clips from 48 classes. It similarly evaluates fine-grained video recognition by having a common diving setting where subtle details of diving sequences define the various categories instead of coarse objects or scenes. We use the standard split containing 16k training/2k test videos. The video lengths have a very wide range: min: 24 frames, max: 822 frames, average: 158 frames. We use the *cleaned-up* labels (denoted 'v2') released in Oct 2020.

Diving48-v2 SotA comparison. We trained publicly available SotA action recognition models on v2 labels, namely: (i) TSM [40], (ii) TSN [59], (iii) TRNms [74], (iv) I3D [5], (v) S3D [66], and (vi) GST-50 [42]. Note, GST-50 is the top-performing method on Diving48-v1 amongst those using a ResNet-50 backbone (refer to v1 comparison in the extended version [72]). The original dataset paper [37] reports results only for TSN and C3D [25]; C3D (2013) is omitted as it is outperformed by more recent methods, *e.g*., I3D and S3D. We could not benchmark against other prominent methods reported for v1, namely, CorrNet [57] and AttnLSTM [26], as their implementation is not public.

Baseline methods. Since we use S3D [66] as the visual backbone for TQN, the following two methods form the baselines: (i) **Short-term S3D (ST-S3D):** following the original dataset papers [37, 46], it is trained on single clips of fixed number (=8 if not specified otherwise) of frames, while for inference, the predicted probabilities from multiple clips spanning a given video are averaged for final classification. (ii) **Long-term S3D (LT-S3D = S3D + Feature Bank):** uses our stochastically updated feature bank at training time in order to pool information from the entire video. Specifically, LT-S3D replicates the ST-S3D's multi-clip evaluation setting at training time, *i.e*., class probabilities obtained from multiple clips spanning the entire video are averaged and used for prediction and backpropagation.

Note at training time ST-S3D incorrectly bases its decisions on *individual* clips which may not contain information relevant for classification. LT-S3D overcomes this issue using multi-clip feature bank and learns better clip-level features (Section 5.2).

Label sets: query-attributes. We define query-attributes independently for each dataset. **Diving48:** The original 48 classes are defined in terms of four *stages* of a diving action. We use four queries corresponding to the stages, and the possible instantiation of each stage as attributes. **FineGym:** Each category in Gym99 and Gym288 is defined by a textual description for the specific sequence *elements* in a gymnastic set, *e.g*., *"double salto backward tucked with 2 twist"*. We extract nouns from these and categorize them into 12 queries, *e.g*., *swing, landing, jump and leap, etc*. and their instantiations form the attributes. Complete factorization is specified in the extended version [72]. In addition to these query and attribute sets, we augment the TQN query set with a "global"query class with the original fine-grained categories as its attribute set, and use its response for the final category prediction.

Metrics. We evaluate on (top-1) classification accuracy, both per original class (48 in Diving48, 99 in Gym99 and 288 in Gym288), and per video.

5. Experiments

5.1. Leveraging Multi-Attribute Labels

We evaluate alternative methods and losses for exploiting multi-part text descriptions on the Diving48-v2 dataset, and compare performance to our *query-attribute* label factorization (Section 3.3). Table 2 summarizes the results for various encoders, decoders and losses; detailed description are given in the extended version [72]. TQN multi-task losses perform the best (81.8% per-video accuracy), followed by

Backbone	**Encoder**	**Decoder** (Aggregation)	**Classification**	**Label**	**Accuracy**	
					per-class	**per-video**
S3D	–	average pooling	multi-class (cross entropy)	class index	72.3	80.4
	self-attention	–			73.7	80.0
		–	multi-label (binary cross entropy)	text descriptions	47.9	50.3
		auto-regressive Transformer	sequence prediction (cross entropy)		51.9	65.1
	–	TQN	multi-task (cross entropy)		**74.5**	**81.8**

Table 2. **Leveraging multi-part text descriptions.** We compare our *query-attribute* label factorization (Section 3.3) to alternative methods for learning with unaligned (no temporal location information) multi-part text descriptions. Our TQN + label factorization outperforms other approaches which are representative of standard classification, and modern encoder-decoder architectures for sequences (see Section 5.1). Evaluation on the Diving48-v2 dataset.

standard multi-class classification (avg. pool: 80.4%, self-attention: 80.0%) which only uses the class index, not the text descriptions. Other sequence based methods for text perform substantially worse ($-$15-30%), due to the restrictive ordering imposed by the text string. TQN goes beyond just attention-based context aggregation, as it outperforms S3D+attention-encoder trained without queries (2nd row) (81.8% vs. 80.4%). This is most likely due to: (i) data re-use enabled by shared sub-labels; and (ii) the learnt queries act as 'experts' to identify discriminative events.

5.2. Feature Bank Ablations

We benchmark our stochastically updated feature banks (Section 3.2) in two ways: first, we evaluate the effect of increasing the temporal context during training, and second the effect of backpropagation through the feature bank. We use Diving48-v2 for both.

Effect of increasing training temporal context. To evaluate the importance of dense and long temporal context during training for fine-grained action recognition, we train the S3D visual encoder [66] on an increasing number of input frames $N = \{8, 32, 64, \text{all frames}\}$, where 'all frames' corresponds to LT-S3D, *i.e.* training with our feature bank. At inference, full temporal support is used for all methods by averaging class probabilities from multiple clips spanning the entire video (no decoder). To control for visual discontinuity between frames due to large input stride, we sample the frames in two ways: (i) *consecutively* sample N frames with a temporal stride of 2 starting at random temporal locations, and (ii) *uniformly* sample N frames with a span of the entire video, where the temporal stride s is proportional to the actual length T of the video, *i.e.*, $s = \lfloor \frac{T}{N} \rfloor$. Table 5 summarizes the results. Consecutive sampling performs better as uniform sampling implies varying input stride which is not amenable to S3D's temporal convolutional filters with fixed stride. More importantly, longer temporal context is better regardless of the frame sampling strategy: all frames: 80.5% per-video accuracy; for $N = \{8, 32, 64\}$: $<$75%. This demonstrates the critical role of our feature bank for training.

Effect of updating bank features during training. A key difference between our feature bank and previous methods for extended temporal support for videos, *e.g.*, *Long-term feature banks* [62] and [34, 44, 52], is that they use *frozen* features, while we continuously update them. We train with frozen/updated features, with/without the TQN decoder on top of visual features, and summarize the results in Table 4. For both with/without TQN, updating the features improves the performance substantially ($\approx +15\%$).

5.3. Comparison with State-of-the-art

Finally, in Table 3 we compare the performance of TQN against SotA methods on Diving48-v2, Gym99 and Gym288. For completeness, performance on the original noisy Diving48-v1 is reported in the extended version [72]. TQN outperforms all methods on all the three benchmarks on both per-video and per-class measures, even when flow+RGB (two-stream) input is allowed for other methods, while only RGB is input to TQN; a detailed table with breakdown for RGB and flow is included in the extended version [72]. Compared to the ST-S3D baseline (S3D with short temporal context), having long-term context (LT-S3D) using our feature bank leads to drastic improvements: $>$30% (absolute) on Diving48-v2, and $>$10% (absolute) on the Gym datasets. Adding TQN decoder on top of LT-S3D leads to further improvements, notably on the 'VT' (vaulting) subset of Gym99 (+5.8%) which contains longer videos: $7-8$ seconds compared to $1-2$ seconds in the 'FX' (floor exercise) subset (+1.2%). Note, the visual backbone of TQN can be made stronger by replacing S3D with, *e.g.*, TSM, TSN, or GST-50. However, we adopt S3D in our experiments as it achieves top performance while fitting within our limited compute budget.

5.4. Performance on Videos of Different Length

In Figure 6 we plot classification accuracy of TQN and the baseline long-term S3D as a function of video length. On videos shorter than 5 seconds, LT-S3D performs similar to TQN as max-pooling suffices to pick out relevant information in short videos. However, TQN's attention based classification outperforms simple pooling for longer videos.

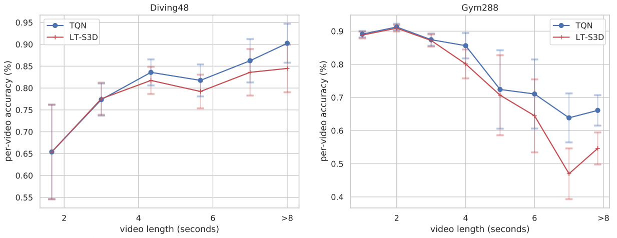

Figure 6. **Classification accuracy on videos of different length.** Mean values plotted with 95% confidence interval.

Network	Pretrained dataset	Modality	# frames in training	**Gym99**				**Gym288**		**Diving48-v2**	
				Per-class	Per-video			per-class	per-video	per-class	per-video
					subset VT	subset FX	total				
I3D	K400	RGB	8	64.4	47.8	60.2	75.6	28.2	66.7	33.2	48.3
TSN	Imagenet	two-stream	3	79.8	47.5	84.6	86	37.6	79.9	34.8	52.5
TSM	Imagenet	two-stream	3	81.2	44.8	84.9	88.4	46.5	83.1	32.7	51.1
TRNms	Imagenet	two-stream	3	80.2	47.3	84.9	87.8	43.3	82.0	54.4	66.0
GST-50	Imagenet	RGB	8	84.6	53.6	84.9	89.5	46.9	83.8	69.5	78.9
ST-S3D	K400	RGB	8	72.9	45.3	82.8	81.5	42.4	75.8	36.3	50.6
LT-S3D	K400	RGB	dense	88.9	69.1	90.4	92.5	57.9	86.3	72.3	80.4
TQN	K400	RGB	dense	**90.6**	**74.9**	**91.6**	**93.8**	**61.9**	**89.6**	**74.5**	**81.8**

Table 3. **Comparison to state-of-the-art.** We compare TQN to several SotA methods for Gym99, Gym288, and Diving48-v2. The results for the Gym datasets are reproduced from the original dataset publication [46], except for S3D [66] and GST-50 [42] was trained by us; no further results are available as the dataset was recently published (2020). For Diving48-v2, since the corrected labels were released recently, we train the publicly available implementations of all methods while replicating the setting of their application to the original Diving48-v1 dataset. TQN achieves top-performance on all three datasets, detailed discussion in Section 5.3.

Description	Visual Encoder	Memory bank	Decoder	Accuracy	
				per-class	per-video
train linear classifier on frozen encoder	frozen	computed offline	avg pool	57.1	66.6
train TQN on frozen encoder			TQN	60.9	68.2
train linear classifier + encoder	fine-tuned	updated online	avg pool	72.3	80.4
train TQN + encoder			TQN	74.5	81.1

Table 4. **Frozen vs. updated feature bank.** To study the importance of updating the feature-bank during training, we train with the TQN decoder and without it (average pooling for temporal aggregation), on top of *frozen* or *stochastically updated* visual features in the memory bank. For both the decoder settings, updating features improves performance drastically ($\approx +15\%$). Evaluation on the Diving48-v2 dataset; details in Section 5.2.

Training				**Test** (w/ all frames)
temporal support	# frames (N)	frame sampling	stride	per-video acc
fixed number of frames	8	consecutive	2	58.8
		uniform	–	50.6
	32	consecutive	2	72.9
		uniform	–	71.2
	64	consecutive	2	74.2
		uniform	–	70.0
full temporal support	proportional to length of videos	memory bank	2	**80.4**

Table 5. **Impact of temporal support during training.** To analyze the importance of temporal support for training and use of stochastically updated feature bank, we train S3D with an increasing number of input frames N and find that longer temporal context consistently improves performance on Diving48.

5.5. Transfer learning with TQN

To investigate the transferability of TQN across domains which differ visually as well as in their query-attributes, we fine-tune the model pre-trained on Gym288 for Diving48. The query vectors $\mathbf{q}$ and the response classifiers $\mathbf{\Psi}$ are tied to dataset specific query-attributes. Hence, to fine-tune on a new dataset, we initialize these randomly and retain the initialization from pre-training for other TQN and visual backbone parameters. We compare this to the random initialization baseline in Table 6. We note that fine-tuning gives better accuracy and trains substantially faster as compared to training from scratch. This is likely because the transformer has learnt (and retained) how to match query vectors to temporal events, and encode the event representation for response classifiers. It is thus able to benefit from the additional training data despite the domain shift.

Pre-training	Epochs	Diving48-v2 top-1
None (random init.)	80	81.8
Gym288	**25**	**83.3**

Table 6. **Transferring TQN.** Per-video performance on Diving48-v2 for a TQN model pre-trained on Gym288.

5.6. Extension to Multi-label Action Recognition

We apply TQN to the Charades dataset [47] and achieves SotA results. Charades labels multiple actions in one video as different classes with precise temporal annotations, as opposed to classification in FineGym and Diving48 where a sequence of combined actions are labelled as one class without localization. Please refer to the extended version [72].

6. Conclusion

We have developed a new video parsing model, the Temporal Query Network (TQN), which learns to answer fine-grained questions about event types and their attributes in untrimmed videos. TQN furthers state-of-the-art in fine-grained video categorization on three datasets, and in addition provides temporal localization and alignment of semantically consistent events. The *query-response* mechanism employed in TQN enables efficient data use through sharing training videos across common sub-labels and outperforms alternative strategies for exploiting textual descriptions. The mechanism is more generally applicable to problems which require spotting entities with varying spans in dense data streams. Our training method with stochastically updated feature banks enables such applications without imposing heavy requirements for expensive large-scale training infrastructure.

Acknowledgements. This research is funded by a Google-DeepMind Graduate Scholarship, a Royal Society Research Professorship, and the EPSRC Programme Grants Seebibyte EP/M013774/1 and VisualAI EP/T028572/1. We thank Weidi Xie, Tengda Han and the reviewers for helpful insights.

References

[1] Anurag Arnab, Chen Sun, Arsha Nagrani, and Cordelia Schmid. Uncertainty-aware weakly supervised action detection from untrimmed videos. In *Proc. ECCV*, 2020. 2

[2] Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proc. CVPR*, 2020. 5

[3] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *Proc. ICCV*, 2015. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, 2020. 2, 3, 4

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 2, 3, 6

[6] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proc. ICML*, 2020. 2

[7] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proc. CVPR*, 2020. 2

[8] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *Proc. NeurIPS*, 2019. 2

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proc. ECCV*, 2018. 2

[10] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*, 2020. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[12] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proc. CVPR*, 2020. 2, 3

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019. 2

[14] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal Transformer for Video Retrieval. In *Proc. ECCV*, 2020. 2

[15] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proc. ICCV*, 2017. 2

[16] Noa Garcia and Yuta Nakashima. Knowledge-based video question answering with unsupervised scene descriptions. In *Proc. ECCV*, 2020. 2, 5

[17] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proc. CVPR*, 2019. 2

[18] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Proc. NeurIPS*, 2014. 2

[19] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. CVPR*, 2018. 2

[20] Meera Hahn, Nataniel Ruiz, Jean Alayrac, Ivan Laptev, and James M Rehg. Learning to localize and align fine-grained actions to sparse instructions. *arXiv preprint arXiv:1809.08381*, 2018. 2

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020. 4

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[23] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Proc. ECCV*, 2016. 2

[24] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proc. CVPR*, 2017. 2

[25] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2012. 6

[26] Gagan Kanojia, Sudhakar Kumawat, and Shanmuganathan Raman. Attentive spatio-temporal representation learning for diving classification. In *Proc. CVPR Workshops*, 2019. 6

[27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 5

[28] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. Progressive attention memory network for movie story question answering. In *Proc. CVPR*, 2019. 2

[29] Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. Multimodal dual attention memory for video story question answering. In *Proc. ECCV*, 2018. 2

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 5

[31] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proc. ICCV*, 2019. 2

[32] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proc. CVPR*, 2014. 2

[33] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 1955. 4

[34] Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, and Shilei Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*, 2017. 2, 5, 7

[35] Xiangpeng Li, Lianli Gao, Xuanhan Wang, Wu Liu, Xing Xu, Heng Tao Shen, and Jingkuan Song. Learnable aggregating net with diversity learning for video question answering.

In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019. 2

[36] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. 2019. 2

[37] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proc. ECCV*, 2018. 2, 5, 6

[38] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G.M. Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41 – 50, 2018. 2

[39] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. Focal visual-text attention for visual question answering. In *Proc. CVPR*, 2018. 2

[40] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. ICCV*, 2019. 2, 6

[41] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *Proc. BMVC*. 2

[42] Chenxu Luo and Alan Yuille. Grouped spatial-temporal aggretation for efficient action recognition. In *Proc. ICCV*, 2019. 6, 8

[43] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proc. CVPR*, 2020. 2

[44] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 2, 5, 7

[45] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proc. ICCV*, 2019. 2

[46] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proc. CVPR*, 2020. 2, 5, 6, 8

[47] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. ECCV*, 2016. 2, 8

[48] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. NeurIPS*, 2014. 2

[49] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[50] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2

[51] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proc. ICCV*, 2019. 2

[52] Yongyi Tang, Xing Zhang, Lin Ma, Jingwen Wang, Shaoxiang Chen, and Yu-Gang Jiang. Non-local netvlad encoding for video classification. In *Proc. ECCV*, 2018. 2, 5, 7

[53] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 4

[54] George Turin. An introduction to matched filters. *IRE transactions on Information theory*, 6(3):311–329, 1960. 1

[55] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018. 2

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017. 1, 2, 3, 5

[57] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proc. CVPR*, 2020. 6

[58] Jinpeng Wang, Yiqi Lin, Andy J. Ma, and Pong C. Yuen. Self-supervised temporal discriminative learning for video representation learning. *arXiv preprint arXiv:2008.02129*, 2020. 4

[59] 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 *Proc. ECCV*, 2016. 2, 6

[60] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. CVPR*, 2018. 2

[61] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proc. ICCV*, 2019. 2

[62] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 2, 5, 7

[63] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware rcnn: A baseline for action detection in videos. In *Proc. ECCV*, 2020. 5

[64] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. CVPR*, 2018. 4

[65] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proc. CVPR*, 2017. 4

[66] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proc. ECCV*, 2018. 2, 3, 5, 6, 7, 8

[67] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proc. ICML*, 2020. 5

[68] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. In *arXiv preprint arXiv:2006.15489*, 2020. 4

[69] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proc. CVPR*, 2020. 2

[70] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-

augmented attention network learning. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017. 2

[71] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. CVPR*, 2015. 2

[72] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. *arXiv preprint*, 2021. 3, 5, 6, 7, 8

[73] Zhou Zhao, Xinghua Jiang, Deng Cai, Jun Xiao, Xiaofei He, and Shiliang Pu. Multi-turn video question answering via multi-stream hierarchical attention context network. In *IJCAI*, 2018. 2

[74] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proc. ECCV*, 2018. 2, 6

[75] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proc. CVPR*, 2020. 2