Hierarchical Conditional Relation Networks for Video Question Answering - Supplementary Material

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran
Applied Artificial Intelligence Institute, Deakin University, Australia
{lethao,vuong.le,svetha.venkatesh,truyen.tran}@deakin.edu.au

1. Introduction

In this supplementary document we include:

- Detailed complexity analysis of CRN units and HCRN models with different hierarchy depths,
- Further implementation details,
- Detail of qualitative analysis and examples (extension of Fig. 1 in the main paper).

2. Complexity Analysis

2.1. CRN units

For clarity, let us recall the notations introduced in our CRN units: k_{max} is maximum subset (also tuple) size considered from a given input array of n objects, subject to $k_{\text{max}} < n$; t is number of subsets randomly selected from sets of all size-k subsets Q^k ($k = 2, 3, ..., k_{\text{max}}$); $g^k(.), h^k(.,.)$ and $p^k(.)$ are sub-networks for relation modeling, conditioning and aggregating, respectively. In our implementation, $g^k(.)$ and $p^k(.)$ are chosen to be set functions and $h^k(.,.)$ is a nonlinear transformation that fuses modalities.

Denoted by F the number of descriptors for each input object for the CRN. Assume that the set function of order k in the CRN's operation in Alg. 1 in the main paper has linear time complexity in k. This holds true for most aggregation functions such as mean, max, sum or product. With the relation orders ranging from $k = 2, 3, ..., k_{\text{max}}$ and sampling frequency t, inference cost in time for a CRN is:

$$C_{\text{CRN}}(t, k_{\text{max}}, F) = \mathcal{O}\left(\frac{t}{2} k_{\text{max}}(k_{\text{max}} - 1)F\right) \qquad (1)$$

The unit produces an output array of size $k_{max} - 1$, each with F features.

2.2. HCRN models

The overall complexity of HCRN depends on design choice for each CRN unit and specific arrangement of CRN units. For clarity, let $t = 2$ and $k_{\text{max}} = n - 1$, which are found to work well in experiments. Let L be the video length, organized into N clips of length T, i.e., $L = NT$.

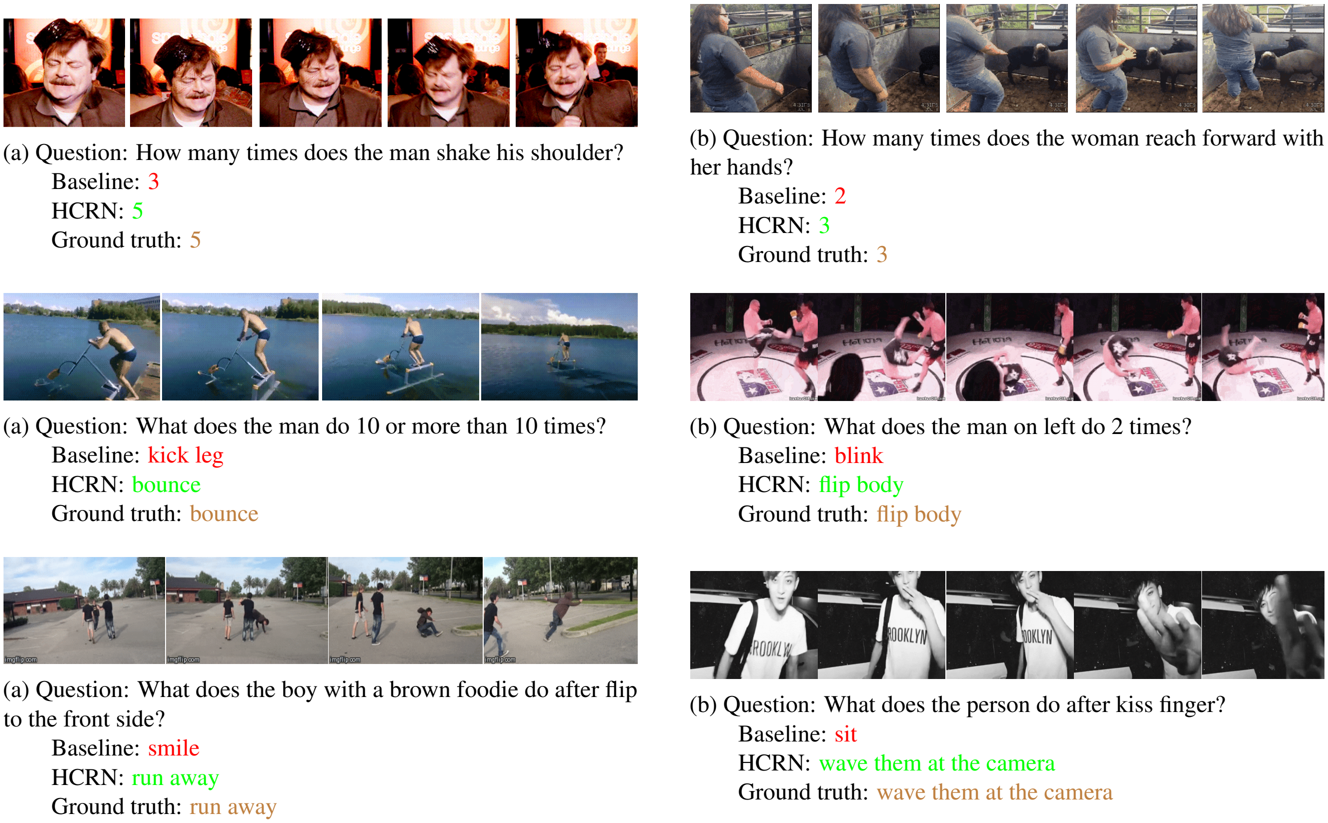
2-level HCRN Consider, for example, the 2-level architecture HCRN, representing clips and video. Each level is a stack of two CRN layers, one for motion conditioning followed by the other for linguistic conditioning. The clip-level CRNs cost $N \times C_{\text{CRN}}(2, T-1, F)$ time for motion conditioning and $N \times C_{\text{CRN}}(2, T-3, F)$ time for question conditioning, where C_{CRN} is the cost estimator in Eq. (1). This adds to roughly $2TLF$ time.

Now the output array of size $(T-4)F$ for the question-conditioned clip-level CRN becomes one in N input objects the video-level CRNs. The video-level CRNs therefore cost $C_{\text{CRN}}(2, N-1, (T-4)F)$ time and $C_{\text{CRN}}(2, N-3, (T-4)F)$ time, respectively, totaling $2N^2TF = 2NLF$ in order. Here we have made use of the identity $L = NT$. The total cost is therefore in the order of $2(T+N)LF$.

3-level HCRN Let us now analyze a 3-level architecture that generalizes the 2-level HCRN. The N clips are organized into M sub-videos, each has Q clips, i.e., $N = MQ$. Since the clip-level CRNs remain the same, the first level costs $2TLF$ time to compute as before. Moving to the next level, each sub-video CRN takes as input an array of length Q, whose elements have size $(T-4)F$. Using the same logic as before, the set of sub-video-level CRNs cost roughly $M \times C_{\text{CRN}}(2, Q-1, (T-4)F)$ time or approximately $2MQ^2TF = 2\frac{N}{M}LF$ (since $N = MQ$ and $L = NT$).

A stack of two sub-video CRNs now produces an output array of size $(Q-4)(T-4)F$, serving as an input object in an array of length M for the video-level CRNs. Thus the video-level CRNs cost roughly $C_{\text{CRN}}(2, M-1, (Q-4)(T-4)F)$ time or approximately $2M^2QTF = 2MLF$ (since $L = NT$ and $N = MQ$). Thus the total cost is in the order of $2(T + \frac{N}{M} + M)LF$.

Recall that the 2-level HCRN has time cost of $2(T + N)LF$. The cost reduction when going from 2-level to



(a) Question: How many times does the man shake his shoulder?
Baseline: 3
HCRN: 5
Ground truth: 5

(b) Question: How many times does the woman reach forward with her hands?
Baseline: 2
HCRN: 3
Ground truth: 3

(a) Question: What does the man do 10 or more than 10 times?
Baseline: kick leg
HCRN: bounce
Ground truth: bounce

(b) Question: What does the man on left do 2 times?
Baseline: blink
HCRN: flip body
Ground truth: flip body

(a) Question: What does the boy with a brown foodie do after flip to the front side?
Baseline: smile
HCRN: run away
Ground truth: run away

(b) Question: What does the person do after kiss finger?
Baseline: sit
HCRN: wave them at the camera
Ground truth: wave them at the camera

Figure 1. Extended examples of hard questions in TGIF-QA that involve combination of motion, near-term and far-term relations.

3-level architectures is $2(N - \frac{N}{M} - M)LF$. Now assuming $N \gg \max\{M, \frac{N}{M}\}$, for example $M \approx \sqrt{N}$ and the number of clips $N > 20$. Then the time saving can be approximated further as $2NLF$. As $N = \frac{L}{T}$, this reduces to $2NLF = 2\frac{L^2}{T}F$. In practice the clip size T is often fixed, thus the saving scales quadratically with video length L, suggesting that going deeper in hierarchy is computational efficient for long videos.

3. Implementation Details

Feature extraction: Our motion feature is extracted based on a pre-trained model of the ResNeXt-101 [3, 1] which results in dividing input videos into N short clips of fixed lengths, 16 frames each. We first locate key frames of N clips depending on the length of a given video and further take 16 consecutive frames in which the corresponding key frames are the central frames of those clips. As videos in the datasets for evaluation, except MSRVTT-QA, are short, we intentionally divide each video into 8 clips (8×16 frames) to produce partially overlapping frames between clips to avoid temporal discontinuity.

Regarding the appearance feature used in the experiments, we take the *pool5* output of ResNet [2] features as feature representation of each frame. This means we completely ignore the 2D structure of spatial information of video frames which is likely to be beneficial for answering Frame QA questions in TGIF-QA dataset. We are aware of this but deliberately opt for light-weighted extracted features, and the main focus of our model is to emphasize the significance of temporal relation, motion, and hierarchy of video data by nature.

Network training: The proposed network is implemented in Python 3.6 with Pytorch 1.2.0. We train the model using Adam optimizer with a batch size of 32. Depending on the amount of training data and hierarchy depth, it may take around 4-30 hours training on one single NVIDIA Tesla V100 GPU.

4. Qualitative Analysis

Fig. 1 extends the qualitative analysis given in Fig. 1 of the main paper. It includes qualitative results of HCRN compared to a baseline on hard cases of four tasks of TGIF-QA. We select the questions that require sophisticated understanding of both near-term and far-term relations in order to give correct answers. We build a baseline method with flat visual-language interaction. Concretely, we use an average pooling over the frame features sequences to obtain the video representation and combine it with the question representation before feeding into a classifier for final prediction. We observed that this baseline struggles in handling those examples. Meanwhile, HCRN shows distinctively power in considering multi-level interaction between motion, question and relations and comes up with good answers.

References

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

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 3

[3] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 3