Supplementary for Dense but Efficient VideoQA for Intricate Compositional Reasoning

This material complements our paper with additional experimental results and their analysis. First of all, we verify the solidity of our model, DSR, with the additional quantitative experimental results in Section 1. Section 2 shows the superiority of DSR in terms of memory efficiency. This is followed by a visual analysis of two modules we proposed, the deformable sampling module and the dependency attention module, in Section 3. Afterward, Section 4 provides some qualitative examples that our model predicts correct answers. Lastly, Section 5 describes the implementation details, such as the settings for training. The code will be made publicly available.

1. Additional Quantitative Results

Table 1 shows the superior performance of our model, DSR, according to the compositional reasoning step of given questions. The compositional reasoning step of a question refers to how many steps the question must be inferred in order to find the correct answer. In other words, the more reasoning step the question has, the more difficult it is. As shown in Table 1, our model consistently performs well compared to the strong baseline, ClipBERT, regardless of the compositional steps. Although our model is designed to target complex questions, it also works effectively for questions with low compositional steps. In particular, there is a large difference in performance for questions with five compositional steps, which require a lot of spatio-temporal reasoning.

		DSR(Ours)	ClipBERT [3]
Step 1	Binary	**75.24**	74.08
	Open	**9.23**	8.46
	All	**74.98**	73.82
Step 2	Binary	**75.99**	75.50
	Open	**46.92**	46.48
	All	**55.87**	55.42
Step 3	Binary	**79.64**	79.61
	Open	**71.24**	70.55
	All	**74.70**	73.87
Step 4	Binary	82.8	**83.82**
	Open	**49.84**	48.86
	All	**54.01**	53.29
Step 5	Binary	**58.34**	48.23
	Open	**57.78**	33.41
	All	**50.26**	38.30

Table 1. Quantitative comparison with ClipBERT on the reasoning step based subset of AGQA dataset.

In addition, we validate that the performance gain of DSR comes from novel modules we proposed, but not from the increased parameters, by ablation for each module considering fair parameter sizes. Based on the ClipBERT architecture that records the best score in Table 4 in the main paper (a), we add 2 transformer layers in place of the dependency attention, for the text embedding (b). In addition to (b), we add 4 transformer layers instead of the conditional sampler, for the visual embedding (c). Remarkably, (b; 52.62) and (c; 52.56) even record lower scores than (a; 53.24). Simply adding extra parameters does not increase the QA accuracy. Also, we postulated that adding question embedding to learnable queries is crucial for deformable sampler. Without conditioning on questions, sampled visual tokens will always become identical no matter which questions are given, which would be suboptimal for complex QA tasks. We got an accuracy of 52.39 without the question conditioning, which is lower than our model. Namely, all the modules we proposed add value.

2. Memory Efficiency of DSR

For the spatio-temporally complex QA task, it is important for a model to cover as many frames as possible efficiently. Here, we compare how many frames can be addressed by each method until the OOM error is raised under the one NVIDIA V100 GPU environment that has 32GB GPU memory. For this experiment, we set the batch size as 1 using the same visual backbone and cross-modal transformer architecture for all methods. For the ClipBERT, each clip consists of 2 consecutive frames, i.e., 64 clips are needed to address 128 frame length, which is the default and the best configuration in their paper [3].

From Table 2, we observe that our DSR shows the

	# of frames								Max
	2	4	8	16	32	64	128	256	Frames
Baseline	7.41	7.61	8.30	10.52	16.08	**OOM**	**OOM**	**OOM**	60
ClipBERT [3]	7.39	7.91	8.86	10.14	12.71	17.84	28.15	**OOM**	162
DSR(Ours)	6.69	6.81	7.21	8.31	9.68	13.64	23.65	32.46	**269**

Table 2. Comparison of memory consumption in GB. Max Frame in the last column means the maximum number of frames right before the OOM error.

best memory efficiency among all comparatives. For the Baseline model, all visual features are fed to the cross-modal transformer without any pooling or sparsification. As a result, the memory requirement of the model increases quadratically according to the length of the full visual feature sequence: $\mathcal{O}((THW + L_t)^2)$ where L_t is the length of question words. In contrast, ClipBERT and DSR can address long sequences more efficiently than the baseline model. In ClipBERT, the feature map is pooled temporally so that only spatial sequence length ($H \times W$) is considered in the cross-modal transformer. However, ClipBERT is less efficient than our DSR. Since the ClipBERT passes multiple short clips independently, the memory requirement of the cross-modal transformer becomes $\mathcal{O}(N_c(HW + L_t)^2)$ where N_c denotes the number of clips. In DSR, the memory consumption is only proportional to $N_q + T$: $\mathcal{O}((N_q + T + L_t)^2)$ where N_q and T indicate the number of learnable queries and the length of global context features, respectively.

3. Visual Analysis on Usefulness of Each Module

As introduced in Section 3 of the main paper, we proposed two novel modules for spatio-temporal reasoning. To provide the qualitative verification on the effectiveness of each module, this section consists of two parts, 1) qualitative examples of sampled visual features in line with the given question, and 2) visualization of dependency attention weights.

3.1. Justification of conditionally sampled visual features

This section justifies the validity of our deformable sampling module, which samples essential visual features conditioned to given questions. Instead of densely sampling redundant visual features, the module samples a few diverse samples, which are especially helpful for answering the given questions. For brevity, Figure 1 and 2 only represents the sampling points on the temporal axis. In Figure 1, we observe that our DSR samples different sets of frames based on each question, which indicates that DSR can sample frames in a question conditional way. Specifically, the model focuses on most of the temporal steps to answer a question in the bottom example of Figure 1, while the top example shows that the model sparsely attends to the specific temporal blocks.

Figure 2 depicts the corresponding video frames along with the sample question. To answer the given question, a model should understand the following actions in chronological order; 1) reading a book, 2) taking a blanket, and 3) snuggling under the blanket. Notably, sampled frames contain all related actions while excluding most of the unnecessary actions.

3.2. Efficacy of dependency attention head

Figure 3 visualizes the first head of the last layer of the dependency encoder we proposed, on the test dataset. The outputs of the dependency encoder turn into the text inputs of the cross-modal transformer for our model, while the baseline models only utilizes pre-trained Bert embeddings [1] as the input. Compared to the Bert embeddings that contain general relationships among text tokens, dependency encoder benefits from simplifying long questions by further embedding hierarchical information.

4. Qualitative Examples of Successful Prediction

In Figure 4, we illustrate the examples of AGQA dataset that our model successfully predicts the answer. As shown in the examples, AGQA dataset consists of problems of solving complex questions for long video sequences containing various actions. For example, in the case of the third row in Figure 4, the problem can be solved only by recognizing the action of wiping glass and walking while understanding the order of the two actions. Therefore, our dense but effective model is needed to understand the comprehensive semantic structures of the video.

5. Implementation Details

This section provides the detailed architecture of our method, including the overall framework and two main modules we introduced. Afterward, we provide the training details, such as hyper-parameters for each objective function, over the dataset we utilized.

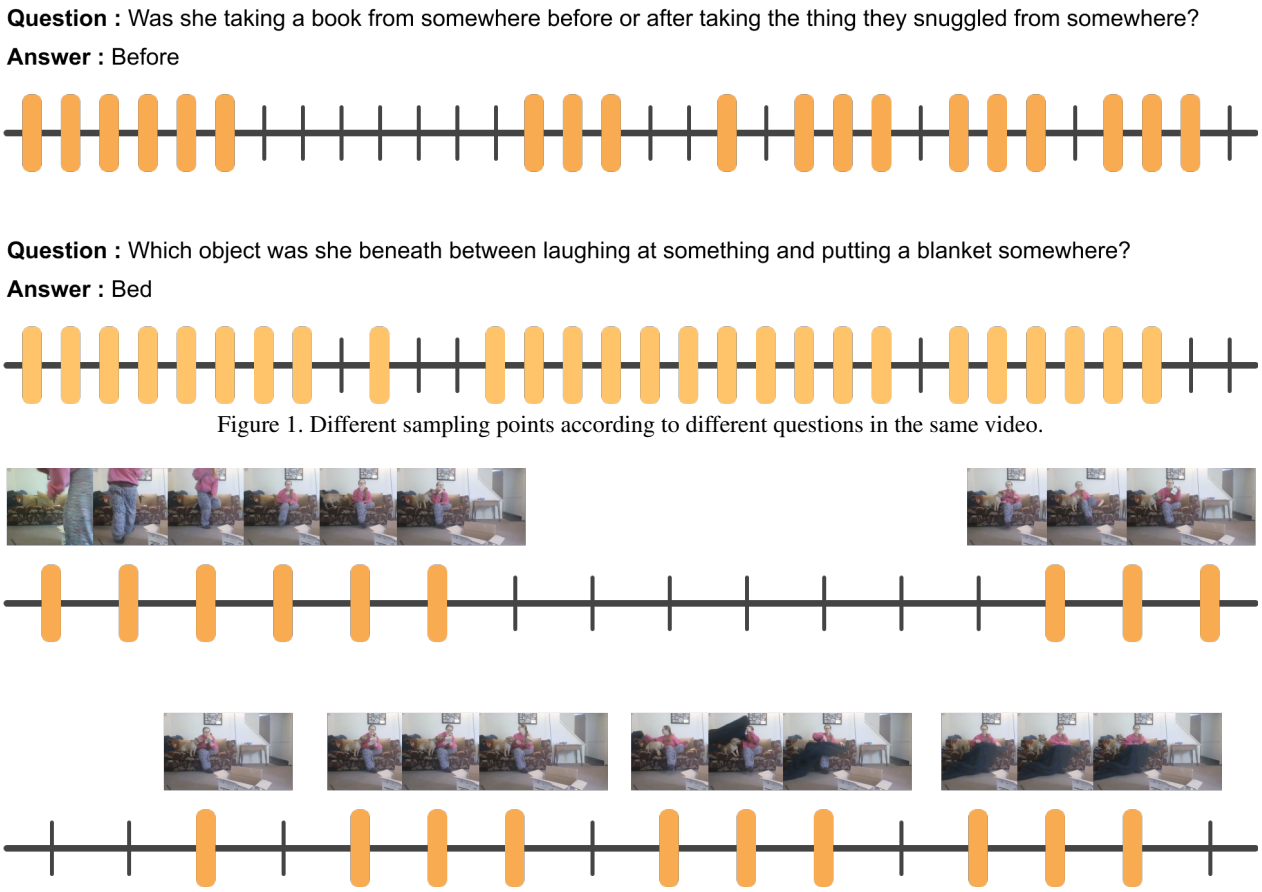

Figure 1. Different sampling points according to different questions in the same video.

Figure 2. Video frames corresponding to the deformable sampling points.

5.1. Model architecture

Transformer based VideoQA model We use the same architecture with ClipBERT [3] for the transformer-based VideoQA model. The number of layers and attention heads of each layer is set to 12. The hidden and intermediate dimensions are 768 and 3072, respectively. Also, we use GELU [2] action function for the transformer layers. For a classification head, we use 2-fully-connected layers.

Conditional Deformable Attention module For the Conditional Deformable Attention (CDA) module, we use a 4-layer transformer based on the deformable decoder [6]. In each layer, we sample 8 offset points with 4 different attention heads. The hidden dimension of each transformer layer is 768 and we use ReLU [5] as an activation function.

Dependency attention module The 2-layer transformer with 12-multi-head is used as a backbone of the dependency attention module. The dependency-constrained attention module is implemented on the first head of the first layer. For the dependency head, attention probabilities are calculated based on dependency parsing relations and learnable value embeddings are multiplied by corresponding attention probabilities.

5.2. Hyperparameters and training details

Default Hyperparameters and Optimization For all experiments by default, we set the learning rate and weight decay for all modules except for the CDA to 5e-5 and 1e-3, respectively. Also, we use AdamW [4] optimizer and use learning rate warmup over the first 10% training steps followed by linear decay to 0. Following the optimization strategy for [6], the base learning rate of CDA module is 0.0001 and the learning rate is decayed at half of the training steps by a factor of 0.1. Also, learning rates of the linear projections, used for predicting object query reference points and sampling offsets, are multiplied by a factor of 0.1. For the video inputs, we resize frames to 448 pixels for the long spatial side and add zero padding to the remaining regions for

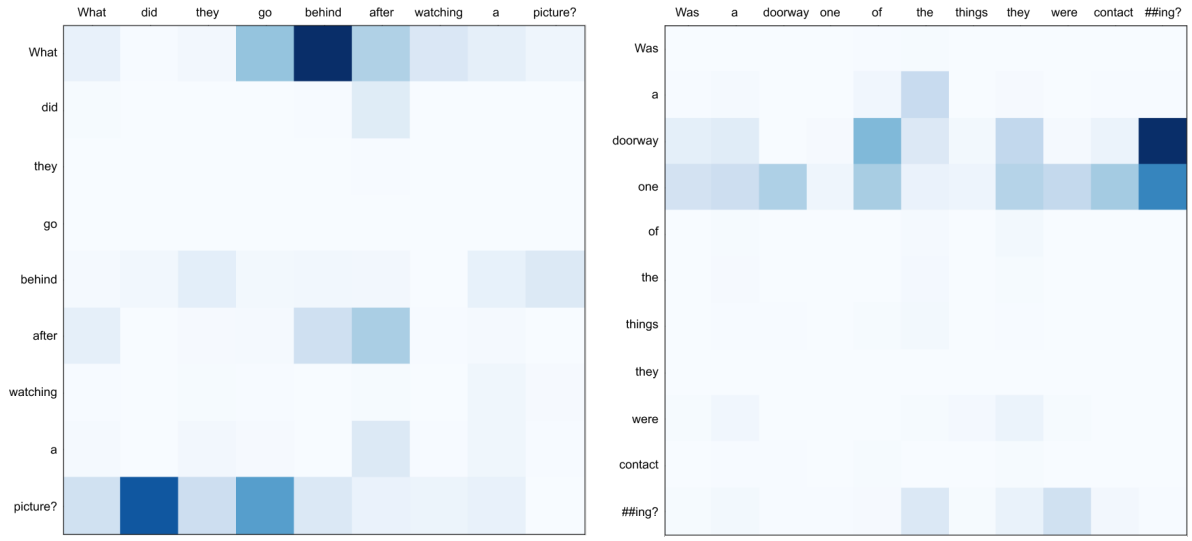

Figure 3. Visualization of self-attention of question tokens that have passed through the dependency attention module.

the short side. Also, we set the maximum question length to 100 for all experiments except for TGIF-QA. For the TGIF-QA the maximum question length is 25.

Training details For our main AGQA experimental results, we train our DSR for 5 epochs with a learning rate of 2e-4. We use a total of 32 NVIDIA V100 GPUs with a batch size of 8 per GPU. For ablations, we use 4 GPUs with a learning rate of 5e-5. For MSRVTT-QA and TVQA, we train our model for ten epochs and five epochs, respectively, and the remaining training details are the same as the main AGQA experiment. For TGIF-QA, we train ClipBERT and DSR for 60 epochs with a dropout probability of 0.4 for the final classification head. Other hyperparameters such as learning rate and weight decay are the same as the default setting described above.

Question : Was a dish one of the things they were contacting?

Answer : No

Question : Had the person ate anything between pouring something into a cup and playing with a phone?

Answer : Yes

Question : Did they walk before or after washing something with the object?

Answer : Before

Qustion : Between holding a dish and putting a dish somewhere, which object did the person grasp?

Answer : Food

Figure 4. Examples of intricate VideoQA problems of AGQA that our model predicts correct answer.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[2] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[3] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learningvia sparse sampling. In *CVPR*, 2021.

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

[5] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[6] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.