Supplementary Material: Learning to Predict Situation Hyper-Graphs for Video Question Answering

In this supplementary document, we discuss the following:

1. Additional architectural details (A)
2. Implementation and training details (B)
3. Additional experimental details (C)
4. Additional results and analyses (D)
5. Qualitative results (E)
6. Computational cost of SHG-VQA (F)
7. Ethical Considerations (G)

A. Additional Architectural Details

In this section, we provide the additional architectural details as follows:

A.1. Input processing

The SHG-VQA can be trained in both open-ended as well as multiple choices settings. For multiple choice setup, C answer choices are also given as input with the question. The goal hence becomes a C-way classification task.

A.2. Question Encoder:

We encode the question (and answer choices) as follows: first, a learnable embedding layer is used to initialize each word token with an embedding vector. These word embeddings along with the special token $[CLS]$ are input to the text transformer encoder encoding each word using multi-head self-attention between different words at each encoder layer. In the multiple choices setup, we append the answer choices to the question words as follows:

$$QA = [CLS] + Q + [SEP] + A0 + [SEP] + A1 + ... + [SEP] + A_C \quad (1)$$

where, QA is the sequence composed of question words and answer choices separated by a special token $[SEP]$; $[CLS]$ is a special token appended at index 0 to aggregate question and answer choices into a sentence vector; C denotes the number of answer choices. We empirically find that composing the question and answer choices in the above format gives better performance.

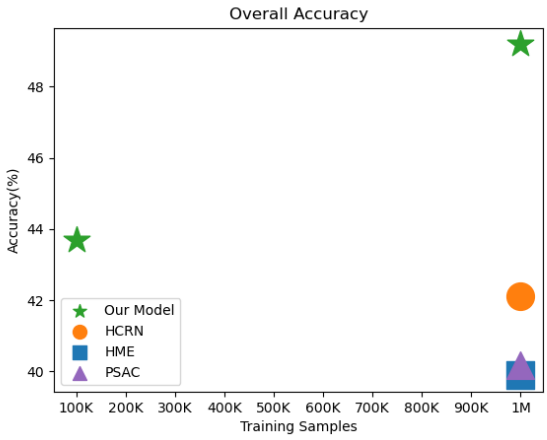


Figure 1. Performance comparison with the baselines w.r.t. training data samples on AGQA dataset. SHG-VQA outperforms the baselines even when trained with only 100K samples.

A.3. Action Decoder

Given the video clip features, we want to decode the set of actions A in each frame. To decode the set of actions A_t at each timestep t, a sequence of learnable embeddings of size d referred to as action queries of length $|N| \times T$ is input to the action decoder. The action decoder comprises the standard transformer decoder architecture and stacks L decoder layers. In addition, the action decoder also takes encoded video tokens as memory and a target mask (of size $\mathbb{R}^{(|N|\times T)\times(|N|\times T)}$). The target mask is created to perform parallel decoding predicting all actions in a frame at once based on the decoded actions in previous frames. The target mask prevents attending to the action queries from future frames (by setting values to $-\infty$) which masks them out [7]. Our approach deviates from using traditional approach [7] or the parallel decoding approach [1] by using a masking at frame-level e.g., to decode actions for frame 0, we set next $|N| \times (T-1)$ positions to $-\infty$ and so on. The action decoder outputs the decoded sequence of action features for each time step t.

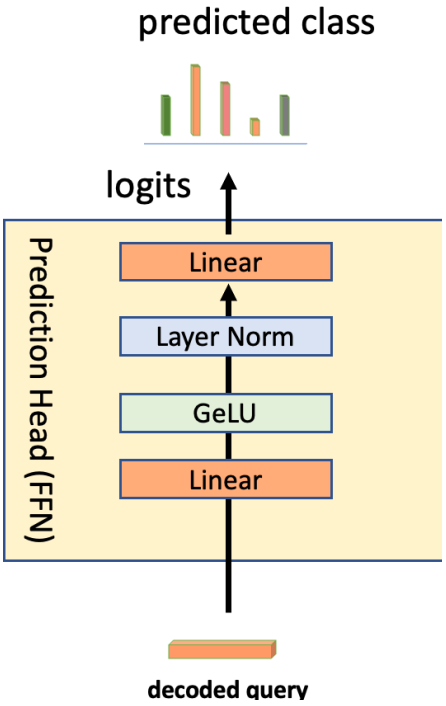


Figure 2. Prediction Head: Each decoded query is passed through a FFN to predict one of the classes or a "no-class" label.

A.4. Relationship Decoder

Like the action decoder, we employ a relationship decoder to decode the set of relations R_t in each frame. We input a sequence of $|M| \times T$ learnable relation embeddings of size d called relation queries and encoded video tokens x_{V_e} as input to the relationship decoder. The relationship decoder has the same architecture as action decoder with variable weights. The relation decoder also takes a target mask (of size $\mathbb{R}^{(|M|\times T)\times(|M|\times T)}$) and perform parallel decoding to predict all relations in a frame at once. The relationships decoder outputs the decoded sequence of relation features for the full video.

A.5. Prediction Heads:

Prediction heads take the decoded queries as input and classify them as an action/relationship from the actual classes or the "no-class" (denoted by ϕ). Therefore, for each prediction head, the total number of classes are $\#classes+1$. See fig. 2 for illustration of prediction head.

B. Additional Implementation Details

B.1. Training details

SHG-VQA is trained with learning rate $lr = 1e-5$, BERT [2] optimizer, and batch size upto 128 for each training depending on the maximum samples which could fit to the GPUs. The best reported results for both datasets use M=8 relations and N=3 actions.

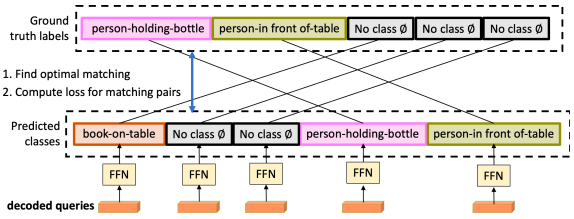


Figure 3. Overview of the bi-partite matching in SHG-VQA. Optimal bipartite matching ($\mathcal{L}_{match}(.)$) is performed between the set of predicted classes for all decoded queries (of actions/relationship predicates) and the ground truth labels using the Hungarian algorithm. Per frame optimal matching is carried $\forall t \in \{1, ..., T\}$. Then, a loss is computed between the matched pairs of ground truth labels and predicted classes using a cross-entropy loss function. See section 3.4 (main paper) for details.

B.2. Hypergraph token handling at test time:

During the training of the situation hyper-graph embedding, we use attention mask for masking padded tokens. However, these masks are not available at inference time because we assume that we are only provided the video and question with answer choices at test time. Thus, we set attention mask to all 1's at inference time. Moreover, the prediction is made to the original video clip with out any data augmentation.

B.3. Data Augmentations

While we obtain a good increase in performance over existing methods for interaction and sequence questions, SHG-VQA performed on par with the baselines for prediction and feasibility. We attribute this to the less training data available for prediction and feasibility. The number of questions for each question type in training set are: interaction:∼16K , sequence: ∼22K, prediction: ∼4K, feasibility: ∼3K. To address this matter, we add training samples from other question types. We filter out the videos from interaction and sequence question types which have the same video ID as prediction and feasibility. This avoids the data leakage problem of peeping into future frames during training. The remaining set of QA pairs are added to the training set for prediction and feasibility resulting in ∼15K training samples for feasibility and ∼16K samples for prediction question type. We observe the expected performance gain when the model is provided proportional amount of data.

C. Additional Experiment Details

C.1. Details about AGQA baselines

For AGQA, we consider three video QA methods as our baselines: PSAC [6], HME [3], and HCRN [5]. PSAC uses

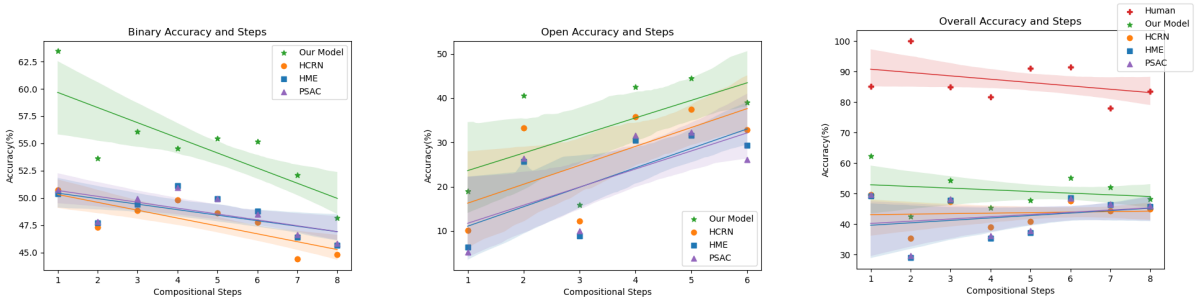


Figure 4. Correlation between accuracy and compositional steps for binary answers, open answers, and overall. To do so, a linear regression model is fit for each model's performance. Our model is superior in performance than the baselines bridging the accuracy gap narrower with the human performance. The shaded area indicates 80% confidence interval.

ResNet-152 to extract video features; HME uses ResNet or VGG for appearance features and use C3D for motion features extraction; HCRN uses ResNet101 for appearance features and ResNext101 pretrained on Kinetics-400 to extract motion features. We use SlowR50 as our backbone model.

C.2. Training Details for STAR Dataset

Data preprocessing for ablation studies: Baselines on STAR dataset train a separate model for each question type. Because training separate models for each question time is not feasible in terms of time and computational resources, we merged the data from all question types for our ablations. To do so, we carefully removed the videos and the corresponding QA pairs from interaction and sequence question types which appear in prediction and feasibility questions. As prediction and feasibility questions are about the future frames not available at inference time, keeping these questions for other question types could give an advantage to the model of looking at the full video even if it happens for solving a different question. Filtering out those QA pairs before merging all questions makes it a fair training for prediction and feasibility questions. However, these questions comprises a large chunk for interaction and sequence. As expected, this declines the VQA performance for interaction and sequence questions upto 2%-8%. However, we notice a gain over prediction and feasibility questions just by showing more examples to the model even if they are not for the same question types. More specifically, on test set, we notice prediction accuracy of 37.29% (merged data) vs. 35.34%(separate) and feasibility accuracy of 33.04% (merged data) vs. 32.52%(separate). We also experimented with using questions from all question types without any filtering and obtained the overall validation accuracy of 48.25%. In Table 8, we provide further details about the experiments including batch sizes for each model, backbone, and loss function. All models were trained up to 100 epochs using early stopping based on the validation accuracy. If not stated otherwise, all ablations are performed with a single model (batch size=32) trained on all questions together with filtering out the QA pairs with overlapped video IDs between {interaction, sequence} and {feasibility, prediction}.

Different batch size per question type/model: As each question type is trained on a separate model with different constrains such as amount of data, but constant hardware requirements, we first evaluate different batch sizes for training each model depending on the maximum number of samples could be used for training. Column 3-*Batch Size* in Table 8 shows a tuple with batch sizes for feasibility, prediction, sequence, and interaction respectively. For training a separate model on each question type, we used batch size=16 with Slow_R50. For ResNext101, we used batch sizes (16, 16, 4, 4) for (feasibility, prediction, sequence, interaction). In our experiments, we observe no significant difference in VQA accuracy when training the models with different batch sizes. Nonetheless, our best results are reported using batch sizes of (16, 16, 16, 16) with Slow_R50 backbone, and batch sizes (16, 16, 4, 4) for ResNext101.

D. Additional results and analyses

Here, we discuss further results and analyses on AGQA and STAR benchmarks.

D.1. AGQA

D.1.1 Performance comparison w.r.t. training data

To train on AGQA, we split the AGQA training set into 90%-10% train-val split. The new training set after this split comprises approximately 1.4M QA pairs. From this training set, we randomly sampled 100K data samples to train our network. We find the SHG-VQA to outperform the baselines even when trained with 100K samples which is $\sim 15\times$ less training data than the data used to train the baseline methods (see fig. 1). More specifically, SHG-VQA obtains 43.69% vs. 42.11% for HCRN which is the best

Table 1. Results on AGQA dataset for different question types w.r.t vision (**w**) and question-only (**w/o**) variants of all models. Best results are shown in **bold** font, second best results are in blue font. SHG-VQA performs better or on par to the baselines with only 100K samples (baselines use 1.6M training samples). Numbers are reported in percentages.

		Reasoning								Semantic			Structure					Overall		
Method		obj-rel	rel-action	obj-action	superlative	sequencing	exists	duration	activity	obj	rel	action	query	compare	choose	logic	verify	binary	open	all
PSAC [6]	w/o	37.91	49.95	50.01	33.59	49.78	50.04	45.77	4.88	38.03	50.04	47.07	31.63	49.57	46.87	50.09	49.97	49.01	31.63	40.26
	w	37.84	49.95	50.00	33.20	49.78	49.94	45.21	4.14	37.97	49.95	46.85	31.63	49.49	46.56	49.96	49.90	48.87	31.63	40.18
HME [3]	w/o	36.44	49.98	50.09	32.53	49.79	50.02	42.67	6.53	36.58	50.05	45.84	29.52	49.16	46.12	50.17	49.93	48.68	29.52	39.03
	w	37.42	49.90	49.97	33.21	49.77	49.96	47.03	5.43	37.55	49.99	47.58	31.01	49.71	46.42	49.87	49.96	48.91	31.01	39.89
HCRN [5]	w/o	37.78	50.12	49.99	33.62	49.78	50.10	43.66	5.15	37.90	50.11	46.22	31.24	49.29	47.36	50.21	50.11	49.12	31.24	40.11
	w	40.33	49.86	49.85	33.55	49.70	50.01	43.84	5.52	40.33	49.96	46.41	36.34	49.22	43.42	50.02	50.01	47.97	36.34	42.11
Ours (100K)	w/o	37.42	49.94	50.06	32.53	49.77	49.97	46.62	5.06	37.57	49.96	47.27	30.92	49.66	46.69	50.01	49.97	48.98	30.92	39.88
	w	41.93	49.26	51.52	35.24	50.11	52.24	45.62	5.61	42.17	51.14	46.36	38.69	49.82	42.37	50.84	52.59	48.77	38.69	43.69
Ours (full)	w/o	38.72	50.03	49.99	33.87	49.85	50.02	48.23	5.80	38.83	50.01	48.11	32.58	49.94	**47.96**	50.16	49.98	49.43	32.58	40.95
	w	**46.42**	**60.67**	**64.63**	**38.83**	**62.17**	**56.06**	**48.15**	**10.12**	**47.61**	**56.19**	**53.83**	**43.42**	**60.68**	47.76	**52.86**	**56.63**	**55.04**	**43.42**	**49.20**

model for AGQA on overall VQA accuracy. Similarly, we obtain on par or often better performance on the three testing metrics of indirect references, novel compositions and more compositional steps. We provide a detailed breakdown of our results with 100K and 1.4M training samples in comparison with the baselines which were trained on the full training set of 1.6M QA pairs. See table 1, 2, 3 for detailed results.

D.1.2 Results for more compositional steps

AGQA provides a train-test split to test models's generalization to more compositional steps where training split has questions with fewer compositional steps. On this metric, SHG-VQA with 100K training samples achieves comparable results to the SOTA model. When compared to the best performing model for each question type, our full model gains ↑ 4.15% absolute points over the best model (HME: 48.09% vs. ours: 52.24%) for binary questions, ↑ 1.2% improvement over SOTA (HCRN:23.70% vs. ours:24.90%) for open-answer questions, achieving overall ↑ 4.14% improvement on all questions. Fig. 4 shows correlation between accuracy and compositional steps. A linear regression model is fit to each method's performance w.r.t number of compositional steps. For **binary questions**, the baseline methods perform significantly lower even with single compositional-step questions, whereas our model unsurprisingly yields the highest accuracy. SHG-VQA is consistently better for all compositional steps on binary question than the baselines. Nonetheless, we observe a negative correlation between accuracy and compositional steps for binary questions. For **open** questions, a slightly positive correlation between accuracy and compositional steps is noticed for all methods including SHG-VQA. For overall accuracy on this metric, although SHG-VQA is able to bridge the gap between human accuracy and VQA algorithms by providing SOTA results, there is still large room for improvement on this novel task.

Table 2. Evaluation on AGQA's **novel compositions.**

Method	training data size	Binary	Open	All
PSAC	1.6M	46.49	19.34	34.71
HME	1.6M	45.42	17.17	33.15
HCRN	1.6M	44.88	20.12	34.13
SHG-VQA	100K	46.55	22.2	36.01
SHG-VQA	1.4M	**49.27**	**25.92**	**39.15**

Table 3. Comparison on AGQA's **more compositional steps** with our model with 100K training samples and full training set.

Method	training data size	Binary	Open	All
PSAC [6]	1.6M	47.65	14.81	47.19
HME [3]	1.6M	48.09	20.98	47.72
HCRN [5]	1.6M	46.96	23.70	46.63
SHG-VQA	100K	47.13	22.66	46.97
SHG-VQA	1.4M	**52.24**	**24.90**	**51.86**

Table 4. Additional results on AGQA for all question types.

	Question Types	Blind Model (Q-Only)	Deaf Model (V+HG)	SHG-VQA-100K
Reasoning	object-relationship	37.42	15.16	**41.93**
	relationship-action	**49.94**	0.01	49.26
	object-action	50.06	0.06	**51.52**
	superlative	32.53	14.88	**35.24**
	sequencing	49.77	0.04	**50.11**
	exists	49.97	17.91	**52.24**
	duration comparison	**46.62**	7.89	45.62
	activity recognition	5.06	0.00	**5.61**
Semantic	object	37.57	13.71	**42.17**
	relationship	49.96	13.92	**51.14**
	action	**47.27**	2.92	46.36
Structure	query	30.92	15.63	**38.69**
	compare	49.66	1.08	**49.82**
	choose	**46.69**	9.72	42.37
	logic	50.01	18.02	**50.84**
	verify	49.97	18.12	**52.59**
	binary	**48.98**	10.65	48.77
	open	30.92	15.63	**38.69**
	all	39.88	13.16	**43.69**

D.1.3 Additional results on AGQA for model variations

AGQA [4] report results for each baseline with language-only model to compare with the respective full models. Following this, we train SHG-VQA in three settings on AGQA:

Table 5. **Results for different training protocols.** Results shown for STAR test set. Rows 1,2, and 3 are with SlowR50 and rows 4,5 show results with MViT-B backbone.

Q. Type	Interaction	Sequence	Prediction	Feasibility	Overall
(1) separate training	**47.98**	42.03	35.34	32.52	**39.47**
(2) all w/ filtered data	37.67	36.91	37.29	**33.04**	36.23
(3) all-SlowR50	42.38	**42.49**	**37.85**	30.78	38.37
(4) all–SlowR50	42.38	42.49	37.85	30.78	38.37
(5) all–MViTB	**43.35**	**44.37**	**38.55**	**33.91**	**40.04**

Table 6. SHG-VQA with SlowR50 backbone evaluated on STAR-Humans. Numbers are reported in percentages.

	Interaction	Sequence	Prediction	Feasibility	Overall
SHG-VQA	**52.00**	**45.00**	31.00	23.00	37.75

blind model (**w/o vision**), deaf model (**vision-only**), and full model. We perform this study using our 100K subset. Results are discussed below: **Blind model performance** We evaluate our question-only model which is a BERT-like 5 layers transformer encoder against our full vision model (table 4) to measure how much linguistic bias our model is able to exploit from the dataset. With results comparable to HCRN's vision and no-vision counterparts, our language model is able to achieve an overall video-question answering accuracy of 39.88%, only 3.81% less than our vision model. The vision model outperforms its language-only counterpart throughout a majority of the question types, however the language-only model has slight improvements over the vision model in duration comparison question types and action semantics, where it performs 1% better. Additionally, the language model also performs slightly better in regards to overall accuracy on binary question types. Further examining binary question categories (table 4) show that the model again performs roughly 1% better than its vision counterpart on binary object-relationship and duration comparison reasoning categories, as well as binary object and action semantic question types. The most noteworthy difference is that this model outperforms the vision model by 4.32% in the choose structural category. Overall, the full model outperforms this language-only model in most categories. **Deaf model performance** In addition to the language-only model, we also train a deaf (vision-only) model to measure biases that may arise from the visual input alone (table 4). This version of our model obtained an overall VQA accuracy of 13.16% on all question types, performing worse than both our full, and language-only models in every question category. From this, we conclude that the visual bias is much less than the language bias.

Ablation on T/M/N? We chose clip length T=16 following prior works. We report results for varying clip length T on AGQA dataset in Tab. 7 with models are trained for 10-15 epochs on 100K QA pairs. Hyperparameters M and N capture the number of actions and relations we want to predict for each frame. Therefore, video length does not effect M/N.

T	binary	open	all
16	48.77	38.69	43.69
24	46.30	38.89	42.57
32	45.50	37.29	41.36

Table 7. Results on AGQA dataset for varying video clip length T.

D.2. STAR

D.2.1 Results w.r.t different training protocols

We experimented with different training protocols for STAR dataset including separate trainings used in [8], single model with filtered questions as explained in C.2, and training a single model on the full training set. We use SlowR50 backbone for this study and find that using separate trainings is most beneficial for interaction questions and overall accuracy. Using filtered questions although perform best for feasibility questions, but it hurts the performance on other question types. When trained on full training data with SlowR50 and MViT-base backbones, using MViT yields better performance.

D.3. Results on STAR-Humans

STAR-Humans is a subset provided by [8] with 400 free-form questions asked by humans. We evaluate SHG-VQA-SlowR50 on STAR-Humans and obtain the results shown in table 6. For this subset, SHG-VQA performs best for interaction questions (52.00%) and worst on feasibility questions (23%).

E. Qualitative results

Comparison between optimal matching with full video compared to optimal matching for each timestep: Figures 5 and 6 show qualitative comparison of predicted situation hyper-graphs from situation hyper-graph decoder in the proposed model. Note that the situation hyper-graph solely relies on the video input. Hence, we show the input video, ground-truth graph, and predicted situation hyper-graphs in two settings: 1) optimal matching with full video instead for each timestep t over L_{Act} and L_{rel} (baseline); 2) optimal matching for each timestep t for the actions and relations set predictions (as described in $eq.2$ and $eq.4$ in the main paper). We observe that when using the optimal matching without imposing the time constraint (i.e., to do optimal matching at each time step), it results in duplicate predictions at frame level. In figure 5 and 6, the first two rows show the video frames and the corresponding ground truth situation hyper-graph respectively. Row 3 shows the situation hyper-graphs we obtain with the optimal matching for

Table 8. Training configurations for SHG-VQA on STAR dataset.

Experiment	backbone	Batch Size	Models trained	Loss	Test set	Overall Acc.
Adapted batch size per QT:						
SHG-VQA (Q + HG)	Slow_R50	(16, 16, 16, 16)	4	$L(eq.1)$	test	39.47
SHG-VQA (Q + HG)	Resnext101	(16, 16, 4, 4)	4	$L(eq.1)$	test	38.28
SHG-VQA (Q + V)	Slow_R50	(8, 8, 8, 8)	4	L_{vqa}	test	30.81
SHG-VQA (Q + HG)	Slow_R50	(16, 16, 16, 16)	4	$L(eq.1)$	test	39.47
SHG-VQA (Q + V + HG)	Slow_R50	(16, 16, 16, 16)	4	$L(eq.1)$	test	39.18
Hypergraph components:						
SHG-VQA (Q + HG) Action only – Act=3	Slow_R50	32	1	$L(eq.1)$	val	38.68
SHG-VQA (Q + HG) Rel. only – Rel=8	Slow_R50	32	1	$L(eq.1)$	val	35.16
SHG-VQA (Q + HG) Both – Act=3, Rel=8	Slow_R50	32	1	$L(eq.1)$	val	39.20
Number of queries						
SHG-VQA (Q + HG) Act=2, Rel=8	Slow_R50	32	1	$L(eq.1)$	val	38.34
SHG-VQA (Q + HG) Act=3, Rel=8	Slow_R50	32	1	$L(eq.1)$	val	39.20
SHG-VQA (Q + HG) Act=4, Rel=8	Slow_R50	32	1	$L(eq.1)$	val	37.06
SHG-VQA (Q + HG) Act=3, Rel=12	Slow_R50	32	1	$L(eq.1)$	val	39.90
SHG-VQA (Q + HG) Act=4, Rel=12	Slow_R50	32	1	$L(eq.1)$	val	38.39

the full video. In edge labels, we show the predicted relationship as well as the count of multiple edges between two nodes. For brevity, we show every 4^{th} frame i.e., frames 1, 5, 9, 13. We can see in row 3, that the predicted graph is sparse and not able to capture all relationships due to suffering from the duplicate predictions problem. Additionally, it sometimes predicts no-class ϕ label i.e. empty set for actions and predictions as we can see in fig 5, row 3, column 3. Row 4 shows the predicted situation hyper-graphs with the imposed constraint of optimal matching at frame-level. We can see that the proposed solution for optimal matching greatly improves the quality of generated hyper-graphs. See Tab. 9 for quantitative results.

Set Pred. Loss	Interaction	Sequence	Prediction	Feasibility	Overall
full video	39.81	40.69	30.17	29.91	35.15
frame-wise	39.42	41.83	33.8	27.48	35.63

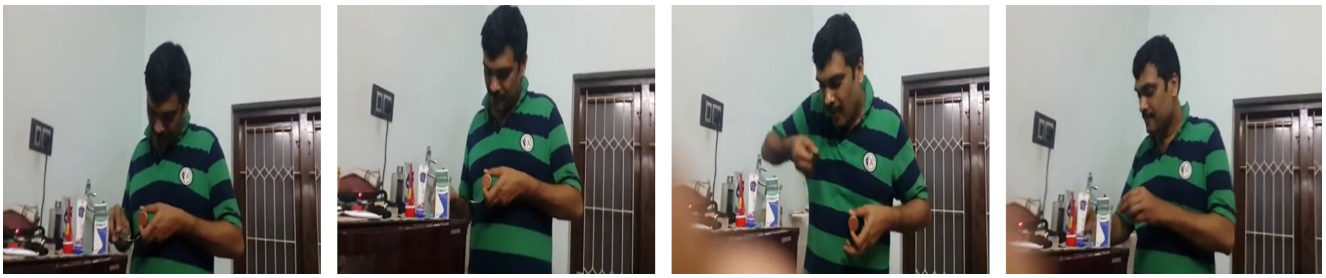
Table 9. Results for SHG-VQA model on STAR test set.

F. Computational Cost of SHG-VQA

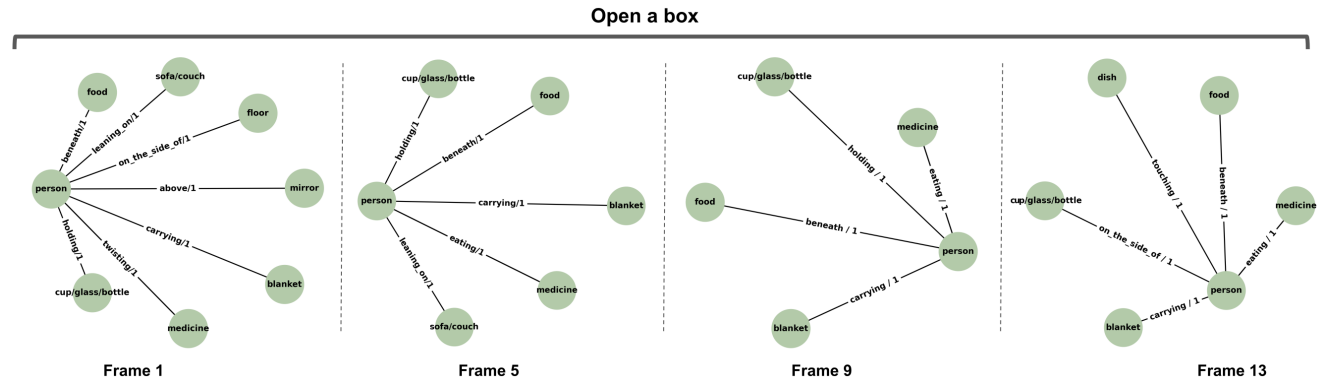
The computational cost of SHG-VQA includes video and text encoders with the little overhead from decoders for decoding graph queries. At inference time, the decoders' output is directly sent to cross-attentional transformer along with the meta embeddings without any graph prediction. Given that we only use L=5 layers for all encoders and decoders in SHG-VQA, the depth of the SHG-VQA is 12 layers i.e., comparable to existing vision-language methods, e.g., ALBEF.

G. Ethical considerations

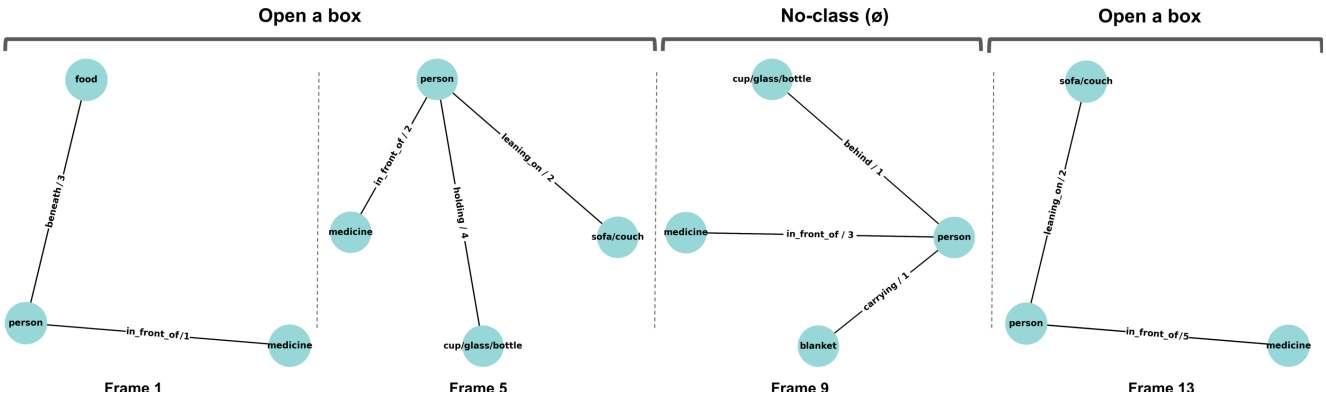
As our system is trained on real-world data, it might capture negative data inherent biases, such as actions only executed by people with specific clothing or stereotype questions. We are not aware of such stereotypes in the here used datasets AGQA and STAR but would recommend assessing the fairness of any system based on this work before putting it in any production environment.



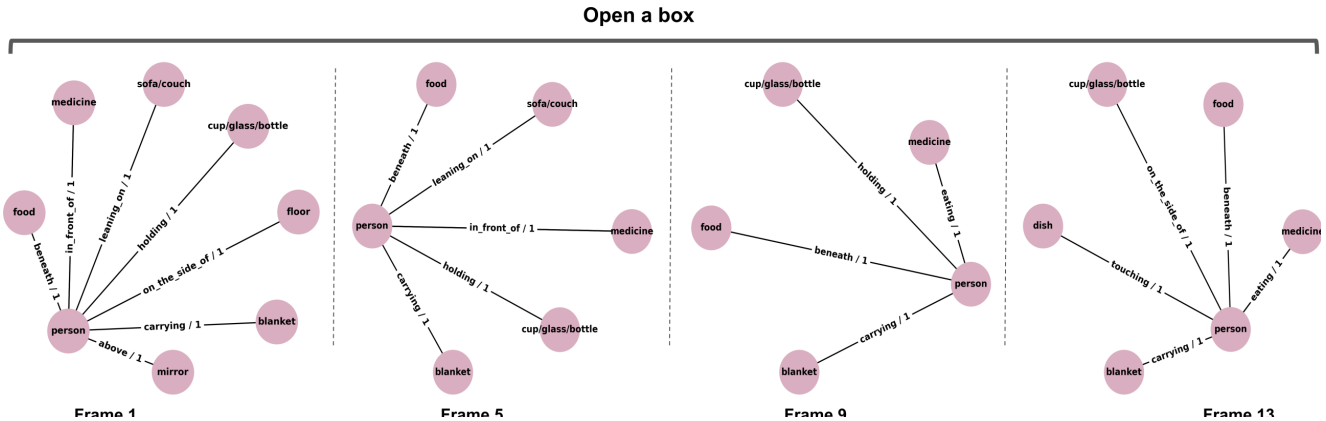
Sample video



Groundtruth hypergraph



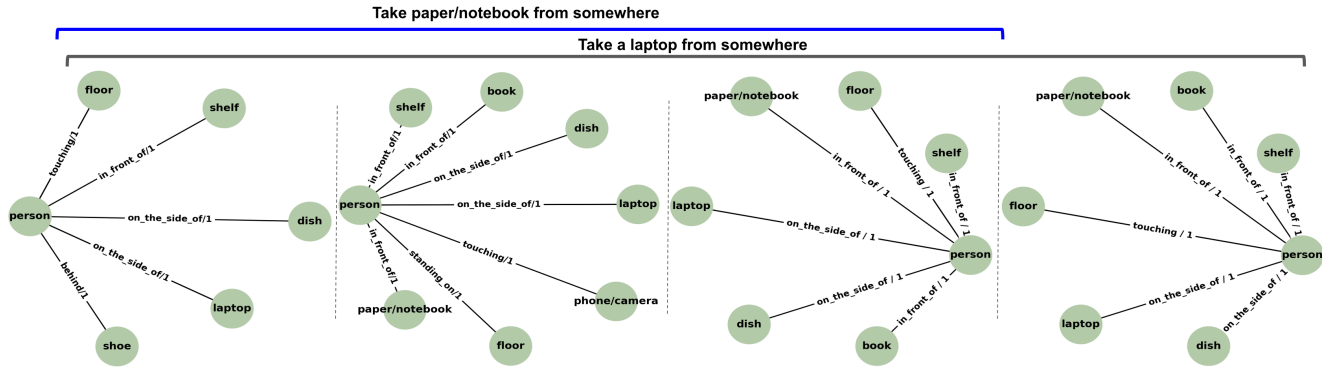
Optimal matching over full video



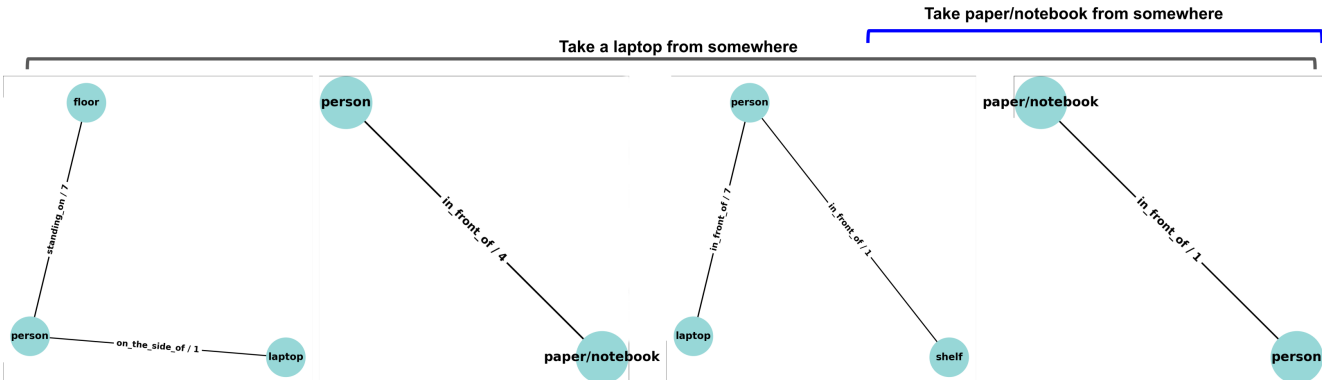
Optimal matching for each timestep t (proposed in eq.2 and eq.4 in the main paper)

Figure 5. Ground-truth and predicted situation hyper-graph for every 4th frame in a clip of length 16. Row 1 shows video frames, row 2 shows the ground-truth situation hyper-graph, row 3 shows predicted graphs from the model with set prediction loss without considering frames, row 4 shows predicted hyper-graph when the model is trained by matching each timestep t (the proposed loss function). The edges show the person-object relationship labels along with the number of times it was predicted. (see Section E for discussion about results.)

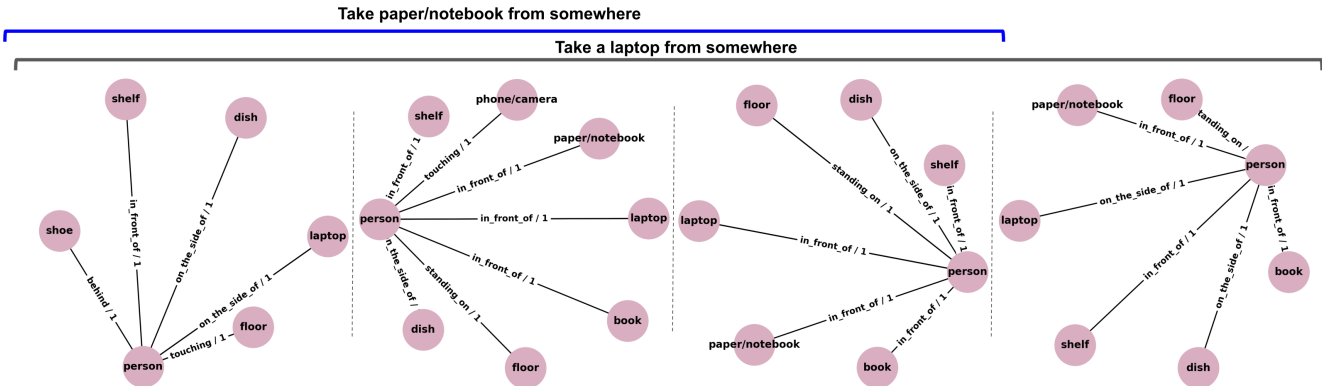
Sample video



Groundtruth hypergraph



Optimal matching over full video



Optimal matching for each timestep t (proposed in eq.2 and eq.4 in the main paper)

Figure 6. Ground-truth and predicted situation hyper-graph for every 4th frame in a clip of length 16. Row 1 shows video frames, row 2 shows the ground-truth situation hyper-graph, row 3 shows predicted graphs from the model with set prediction loss without considering frames, row 4 shows predicted hyper-graph when the model is trained by matching each timestep t (the proposed loss function). The edges show the person-object relationship labels along with the number of times it was predicted. (see Section E for discussion about results.)

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[3] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. *CoRR*, abs/1904.04357, 2019. 2, 4

[4] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297, 2021. 4

[5] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. *CoRR*, abs/2002.10698, 2020. 2, 4

[6] 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. In *AAAI*, 2019. 2, 4

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[8] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. In *Thirty-fifth Conference on Neural Information Processing Systems*, 2021. 5