Supplementary Material for EgoTaskQA: Understanding Human Tasks in Egocentric Videos

Baoxiong Jia[1,2†]
baoxiongjia@ucla.edu

Ting Lei[2,3†]
ting_lei@pku.edu.cn

Song-Chun Zhu[2,3,4]
sczhu@bigai.ai

Siyuan Huang[2]
syhuang@bigai.ai

[1]UCLA Center for Vision, Cognition, Learning, and Autonomy (VCLA)
[2] Beijing Institute for General Artificial Intelligence (BIGAI)
[3] Institute for Artificial Intelligence, Peking University
[4] Department of Automation, Tsinghua University
https://sites.google.com/view/egotaskqa

A Data Collection

A.1 Data Statistics

In this section, we provide more details and statistics on the annotated data. We leverage the compositional action annotation provided in LEMMA [1] and use objects in all semantic positions as the initial set of interacting objects. We remove temporally static, *i.e*. not changing, objects like sofa, floor, hand, table and use them as general reference in action annotation (*e.g*. we consider only "bottled-water" for action "get bottled-water from table using hand" and use "get something from table" for action reference). We annotate states as well as spatial relationships of these objects for both before and after actions and obtain a total of 30K before-after pairs over the 10K action segments annotated in LEMMA. Moreover, we annotate the spatial relationships of the actor, his acting relationship (*e.g*. "get meat using fork" indicates ⟨P1⟩[getting]⟨meat⟩ and ⟨P1⟩[getting-with]⟨fork⟩), as well as his multi-agent relationships annotated as discussed in Sec. 3.1.

We visualize the statistics of annotated relationships in Fig. 1. As we can see from the histogram, spatial relationships of objects were annotated the most, followed by multi-agent relationships like "aware of others" and "looking at". This meets our expectation of the frequent changes in objects' spatial relationships during goal-oriented task execution. Action-related relationships also make up a considerable portion of overall relationship annotations and describe detailed relationships between the person and the target object (*e.g*. getting, putting, pouring) or the tool object (*e.g*. getting-with, cutting-with, putting-with). We visualize the statistics of relationship pairs in Fig. 3.

We list all annotated object attributes and their state values in Tab. 1, and visualize their statistics in Fig. 2. We add an option "unknown" to all attributes for annotating unclear scenarios and ignore this answer during question generation. As shown in Tab. 1 and Fig. 2, we consider various time-varying object attributes including visibility, affordance (*e.g*. cuttability, edibility), and task-dependent status (*e.g*. emptiness, shape). In Fig. 2 (right), we plot the number of changes for each object attribute. In addition to spatial relationship changes described previously, there is an increasing number of occurrences from affordance changes to visibility changes and, finally, task-dependent status changes.

[†]Work done during internship at BIGAI.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

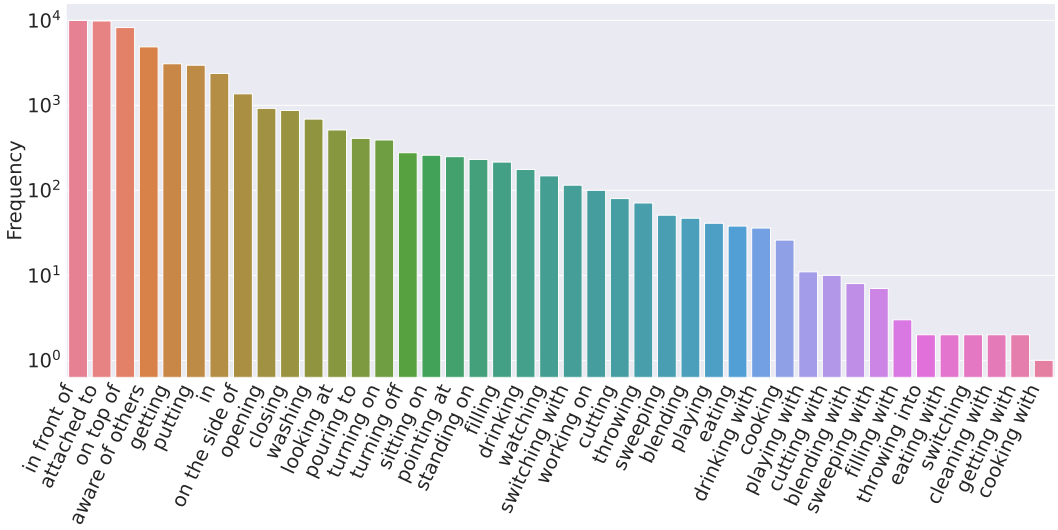

Figure 1: Statistics of relationships annotated during EgoTaskQA data collection. Frequencies are normalized to log-scale for better visualization.

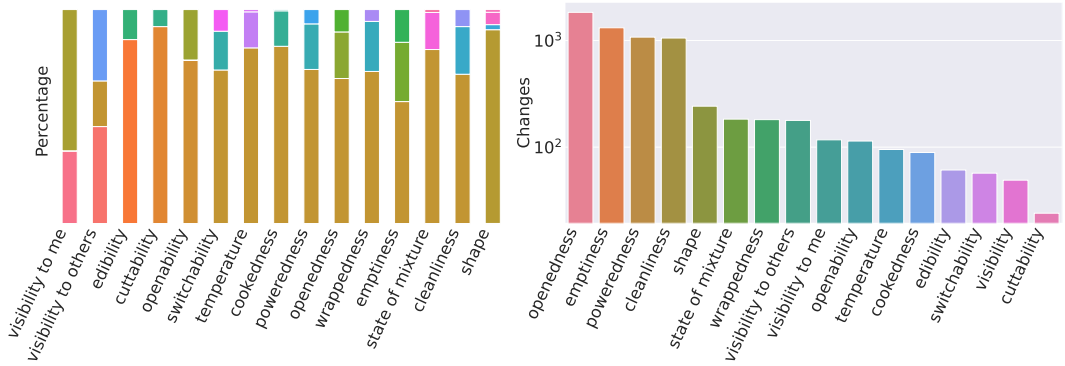

Figure 2: Statistics of object state/attribute change: (left) the ratio of each annotated attribute value for its corresponding object attribute (*e.g.* visible and invisible for "visibility to me"); (right) the frequency of object attribute that was changed by actions (normalized to log-scale for better visualization).

As LEMMA is recorded in indoor environments (kitchens and living rooms), we also observe a large amount of containment relationship changes ("open/close" and "emptiness"). We argue that this data is also potentially beneficial for the study of containment relationships [2].

Table 1: A full list of time-varying object attributes considered and their corresponding possible values.

Attribute	Type	Possible State Values
visibility to me	visibility	visible to me / invisible to me / unknown
visibility to the other person	visibility	visible to the other person / invisible to the other person / unknown
edibility	affordance	edible / can not be eaten / unknown
cuttability	affordance	cuttable / not cuttable / unknown
openability	affordance	openable / can not be opened / unknown
switchability	affordance	can be turned on / can not be turned on / unknown
temperature	status	boiled / in room temperature / unknown
poweredness	status	on / off / unknown
cookedness	status	cooked / raw / unknown
wrappedness	status	wrapped / unwrapped / unknown
emptiness	status	empty / full / unknown
state of mixture	status	mixing / not mixing / unknown
cleanliness	status	clean / dirty / unknown
shape	status	whole / part / diced / fluid / unknown

A.2 Causal Dependency

We describe the details for determining the causal relationship between actions here. We adopt rules for deciding the causal dependency between actions. Given two actions a_1 and a_2, and their state annotations s_1 and s_2, we determine the causal dependency between them as shown in Algorithm 1.

We first collect all interactive object set O_1 and O_2 for a_1 and a_2, and see if there exists an overlap of objects. If no, we assume a_1 and a_2 is not *related*. Next, for each object o that is interacted in both actions, we check whether a_1 lead to the change of attribute s, which is a precondition of a_2's change on o. This condition is validated by checking if o changed the same attribute s in both a_1 and a_2, and the status after a_1 equals the status before a_2, *i.e.* $s_{1,o}^{\text{after}} = s_{2,o}^{\text{before}}$. We say that a_1 and a_2 are causally *dependent* if this condition is satisfied. If there exists an attribute s that was affected by a_1 and did not change during a_2, *i.e.* $(s_{1,o}^{\text{before}} \neq s_{1,o}^{\text{after}}) \wedge (s_{1,o}^{\text{after}} = s_{2,o}^{\text{before}})$, we say that these two actions are *related* since we can not determine whether this relationship is causal or not from the annotations. As currently we did not use additional human resources for verifying each of this *related* actions, we limit our scope of question generation to the *dependent* and *unrelated* action pairs. After checking the causal dependency for all action pairs in the video, we recursively construct the dependency tree by taking each action as root and adding actions that are dependent on all dependants of the action to its dependants set. During the recursion, we update the dependency for a newly added action to *related* if there exist *related* dependency relationships in the path from the root action to it.

Algorithm 1: Causal Dependency Check

Input: two actions a_1 and a_2 and their object state annotation S_1 and S_2.
Output: the causal dependency relationships between a_1 and a_2.

Gather all interactive objects $O_1 = \{o_i^1\}_{i=1}^m$ and $O_2 = \{o_i^2\}_{i=1}^n$ in action a_1 and a_2.
if $O_1 \cap O_2 = \varnothing$ **then**
  return *unrelated*
else for $o \in O_1 \cap O_2$ **do**
  for $s_{1,o} \in S_1$, $s_{2,o} \in S_2$ **do**
    if $(s_{1,o}^{before} \neq s_{1,o}^{after}) \wedge (s_{1,o}^{after} = s_{2,o}^{before}) \wedge (s_{2,o}^{before} \neq s_{2,o}^{after})$ **then**
      return *dependent*
    else if $(s_{1,o}^{before} \neq s_{1,o}^{after}) \wedge (s_{1,o}^{after} = s_{2,o}^{before})$ **then return** *related*
return *unrelated*

B Question-Answer Generation

B.1 Preprocessing Annotations

To generate answers, we first collect video intervals as mentioned in Sec. 3.1. These clips are cropped from original videos to contain 4∼5 actions on average. We further concatenate the next three actions performed by the actor that is unseen in videos into the intervals of interest for generating predictive questions. After generating these intervals of interest, we gather all corresponding annotations, including annotations for both the actor's action and the helper's (*i.e.* the other person's) simultaneous actions. We organize these annotations in a dictionary for convenience purposes.

B.2 Operator and Program Design

We use a template-based method for generating questions. More specifically, we design operators that work on the annotation dictionaries with different purposes. Inspired by previous works [3–5], we design nine basic operators for composing the logic for each program template. We provide the specification of each operator, and its usage with an example, in Tab. 2. The basis of these programs lies in the conditional query, similar to database queries. We use $A@B$ for filtering data with the attribute A equal B, and we use $A\$$ for querying the value of attribute A from data.

We provide the full list of program templates in Tab. 4. In these templates, we use $\{a\}$ for representing the parameter type action, $\{o\}$ for the parameter type object, $\{f\}$ for object attributes and $\{fv\}$ for attribute values. During the question generation process, we substitute these positional arguments with the corresponding sample space to initialize these program templates so that the resulting programs are executable on the annotation dictionary. If the program becomes not executable at any intermediate step, we return `None` for the corresponding operator to stop the trial, reinitialize the template and make a new attempt. This generation process concludes to 368K question-answer pairs.

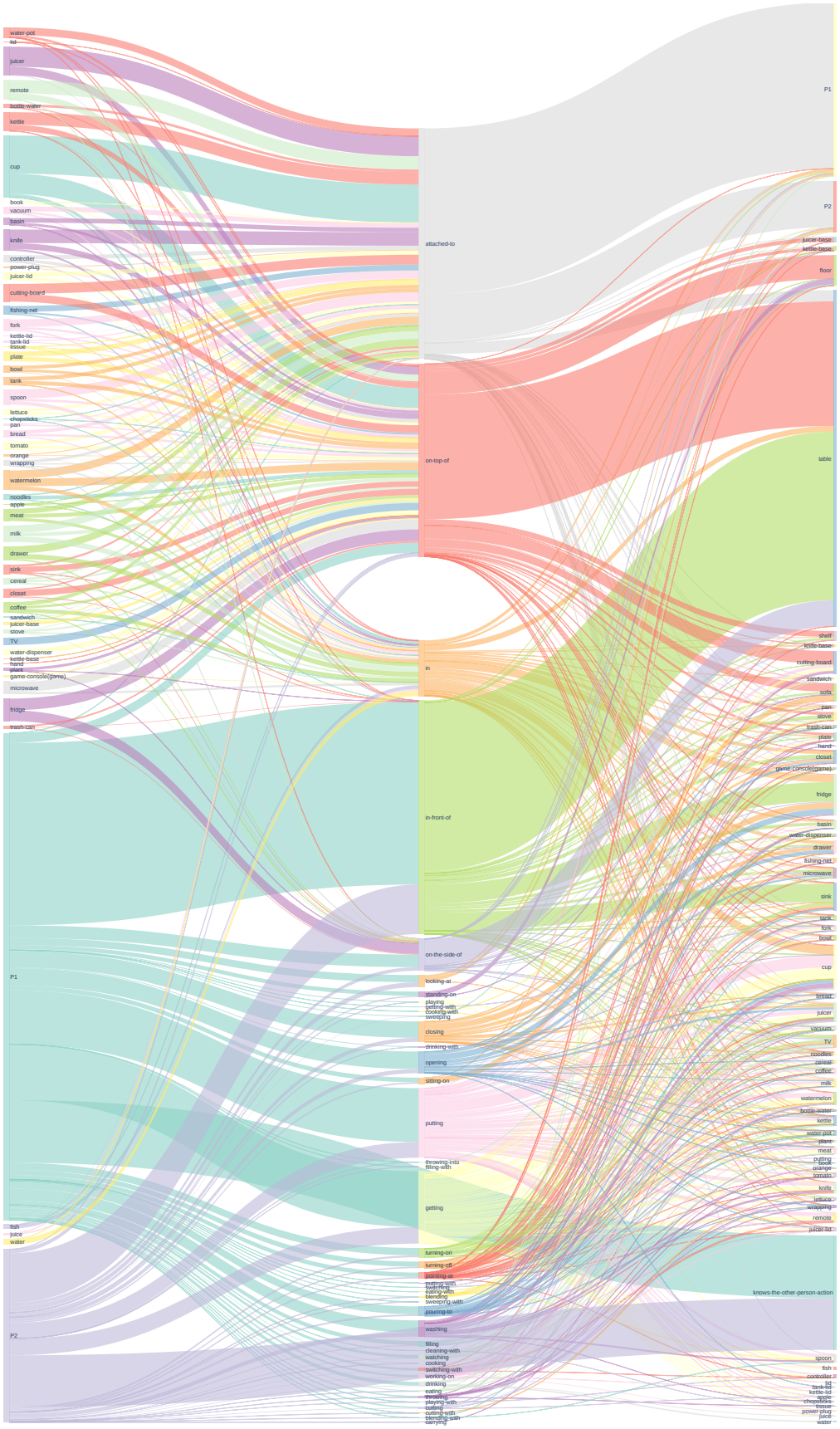

Figure 3: Visualization of all relationship pairs and their corresponding ratios.

Table 2: All program modules used for question-answer generation.

Query Operation	Parameter List	Return Type	Usage and Example
Filter	arg_1: conditions arg_2: intervals	intervals	Return the intervals that satisfies the conditions. `filter([obj@spoon, change@cleanliness], video)`
Only	arg_1: intervals	interval	Return the only interval from list, return `None` if $\|arg_1\| \neq 1$. `only(filter([action@putting]))`
Localize	arg_1: before/after arg_2: interval	intervals	Return all intervals before/after the interval provided in arg_2. `localize(before, only(filter(action@putting)))`
IterateUntil	arg_1:forward/backward arg_2: intervals	interval	Return the first interval of the interval list from the front/back. `iterate_until(forward, filter([change@emptiness], video))`
Query	arg_1: conditions arg_2: interval	value	Return the value from the interval identified by the conditions. `query([aware@yes, action$], only(filter([action@getting], video)))`
Verify	arg_1: conditions arg_2: interval	bool	Verify arg_1 in the interval arg_2, return "yes" if satisfied, "no" otherwise. `verify([change@openedness, obj@closet], only(filter([action@closing], video)))`
Pred	arg_1: intervals	intervals	Return the anticipating intervals within intervals arg_1 `pred(filter([action@pouring], video))`
Counterfactual	arg_1: conditions arg_2:intervals	intervals	Return the original intervals with executability of each interval adjusted according to the counterfactual query arg_2. `counterfactual([action@getting], video)`
Depend	arg_1: interval arg_2: interval	bool	Return "yes" if interval arg_1 and $interval_2$ are dependent, "no" otherwise. `depend(only(filter([action@opening], video)), only(filter([action@closing], video)))`

Table 3: Question-answer pair statistics before and after balancing.

	World	Intent	Multi-agent	Descriptive	Predictive	Counterfactual	Explanatory	Action	Object	State	Change	Open	Binary
Before	299K	43K	53K	181K	22K	71K	102K	122K	14K	105K	126K	182K	186K
After	32K	5K	6K	21K	4K	6K	9K	17K	4K	9K	10K	26K	13K

For indirect questions, we make use of the provided templates and use indirect reference to substitute parameters for action ($\{a\}$) and object ($\{o\}$). We list all indirect reference considered in Tab. 5. In our experiments, we consider simple temporal references of actions, including "before some action", "after some action", "the first action" and "the last action". For object references, we consider using state change for referring objects. We use "the first object that has status change", "the last object that has status change" as well as "the object that has status change" to refer to objects. Concretely, we substitute the corresponding positional arguments $\{a\}$ and $\{o\}$ shown in Tab. 4 with the substituting text and program templates shown in in Tab. 5. We use the template with $\langle$ and $\rangle$ to substitute the action/object positional arguments in the original program template for generating programs. As this substitution could be easily adapted to have multi-step indirect references, we limit the indirect references in our benchmark to 1-step indirect references to avoid generating questions that are difficult to understand. To facilitate models' understanding of indirect references, we add additional questions on these indirect queries for objects and actions. We leverage the program template shown in Tab. 5 and adjust the text to have questions like "what is the first action...", "what is the action before/after...", and "which object changed its status first..".

B.3 Question-answer statistics and balancing

In this section, we provide the details for balancing and question-answer statistics. As described in Sec. 3.2, we balance the questions according to their reasoning types and obtain 40K diverse question-answer pairs. We visualize the most common question texts in Fig. 4. For balancing, we follow the algorithm provided by [5] and adjust the open-answer problems to ensure that the top 20% answers of each reasoning type do not answer to more than 33% questions in the same type. We select this ratio to get a smoother answer distribution while not deleting too many questions in the whole set. To avoid overfitting to the binary answer distribution, we control the ratio between open-answer and binary questions to be 2:1. We show the statistics for each general question type before and after balancing in Tab. 3.

C Experiment

In this section, we provide details on model implementation, hyper-parameters selection, and environment setup. We provide the details for each evaluated baseline model as follows:

- **VisualBERT**: We use the pretrained VisualBERT model and implementation provided by Hugging Face [6]. Specifically, we uniformly sample 20 frames per video and extract visual features using ResNet to generate visual tokens. We use the pretrained BertTokenizer for embedding text tokens. To avoid instabilities during training, we set the learning rate to 5×10^{-5}. As we observe convergence at around 35 epochs, we set the total training epochs to be 40 with a batch size of 32. For its language-only variant, BERT, we adopt the same setting and train for 25 epochs as we observe faster convergence compared to its vision-language counterpart.

Table 4: Question templates adopted in EgoTaskQA. We use "obj" short for "object" and "hoi" short for "human-object interaction".

Template	Program	Type	Scope	Semantic	Overall
which object changed its status when the person {a}?	query([change, obj$], only(filter([action@{a}], video)))	descriptive	world	object	query
what status of {o} changed while the person {a}?	query([change, obj@{o}, change$], only(filter([action@{a}], video)))	descriptive	world	change	query
Is {o} visible to the other person {t} the person {a}?	query([state, type@{t}, obj@{o}, 'visibility to the other person'$], only(filter([action@{a},is_multi@yes], video)))	descriptive	world, multi-agent	state	verify
Is the other person aware when the person {a}?	query([aware], only(filter([others], filter([action@{a}], video))))	descriptive	multi-agent	action	verify
what is the other person doing while the person {a}?	query([hoi], only(filter([aware@yes, others], filter([action@{a}], video))))	descriptive	multi-agent	action	query
which object changed its status when the other person {a}?	query([change, obj$], only(filter([action@{a}], filter([aware@yes, others], video))))	descriptive	world, multi-agent	object	query
What is the status of {o} {t} the person {a} to change it?	query([change, obj@{o}, {t}$], only(filter([action@{a}], video)))	descriptive	world	state	query
what is the status of {o} {t} the other person {a} to change it?	query([change, obj@{o}, {t}$], only(filter([action@{a}], filter([aware@yes, others], video))))	descriptive	world, multi-agent	state	query
did the attribute of {o} changed because of the person's action {a}?	verify([change, obj@{o}], only(filter([action@{a}], video)))	descriptive	world	change	verify
during which action does the person knows about the other person's action?	query([hoi], only(filter([aware@yes], video)))	descriptive	multi-agent	action	query
what will the person want to have {o}'s {f} be in the future?	query([change, obj@{o}, change@{f}, after$], only(filter([change, obj@{o}, change@{f}], pred(video))))	predictive	intent	state	query
what status will the person change on {o}?	query([change, obj@{o}, change$], only(filter([change, obj@{o}], pred(video))))	predictive	intent	change	query
what does the other person want to have the {f} of {o} be?	query([change, obj@{o}, change@{f}, after$], only(filter([change, obj@{o}, change@{f}], pred(filter([others], video)))))	predictive	intent	state	query
what will the other person change on {o}?	query([change, obj@{o}, change$], only(filter([change, obj@{o}], pred(filter([others], all)))))	predictive	intent	state	query
what will the person do next after this video?	query([hoi], iterate_until(forward, pred(video)))	predictive	intent	action	query
what will the other person do next?	query([hoi], iterate_until(forward, filter([aware@yes, others], pred(video))))	predictive	intent, multi-agent	action	query
will {o} be visible to the other person after the person's next action?	query([state, type@after, obj@{o}, 'visibility to the other person'$], iterate_until(forward, filter([is_multi@yes], pred(video))))	predictive	world, intent, multiagent	state	verify
if the actor do not {a}, which object will he/she not be able to change in the future?	query([change, obj$], only(filter([executable@no], filter([change], pred(counterfactual([action@{a}], all)))))))	predictive, counterfactual	world	object	verify
what will the status of {o} change to if the actor {a} in the future?	query([change, obj@{o}, after$], only(filter([action@{a}], pred(video))))	predictive, counterfactual	world	state	query
which object's status is possible to be changed in the future?	query([change, obj$], only(filter([change], pred(video))))	predictive	world	object	query
if the person did not {a_1}, is the person able to {a_2}?	query([executable], only(filter([action@{a_2}], counterfactual([action@{a_1}], all))))	counterfactual	world	action	verify
if the person did not {a}, what remaining actions in the video is executable?	query([hoi], only(filter([executable@yes], counterfactual([action@{a}], all))))	counterfactual	world	action	query
if the person did not {a}, what remaining actions in the video is not executable?	query([hoi], only(filter([executable@no], counterfactual([action@{a}], all))))	counterfactual	world	action	query
if the other person did not {a_1}, is the person able to {a_2}?	query([executable], only(filter([action@{a_2}], counterfactual([others, action@{a_1}], all))))	counterfactual	multi-agent	action	verify
if the other person did not {a}, what actions of this person in the video is executable?	query([hoi], only(filter([executable@yes], counterfactual([others, action@{a}], all))))	counterfactual	multi-agent	action	query
if the other person did not {a}, what actions of this person in the video is not executable?	query([hoi], only(filter([executable@no], counterfactual([others, action@{a}], all))))	counterfactual	multi-agent	action	query
if the person did not {a}, will {o} change its status?	query([executable], only(filter([change, obj@{o}], counterfactual([action@{a}], all))))	counterfactual	world	change	verify
what does the person want {o} to be for doing the action {a} in the video?	query([change, obj@{o},after$], only(filter([action@{a}], video)))	explanatory	intent	state	query
which attribute does the person want to change with {o} for doing the action {a} in the video?	query([change, obj@{o}, change$], only(filter([action@{a}], video)))	explanatory	intent	change	query
how did the person changed the {f} of {o}?	query([hoi], only(filter([change, obj@{o}, change@{f}], video)))	explanatory	world	action	query
what action caused {o}'s status to change to {fv}?	query([hoi], only(filter([change, obj@{o}, change@{f}, after@{fv}], video)))	explanatory	world	action	query
does action {a_1} fulfills the preconditions of action {a_2}?	depend(only(filter([action@{a_1}], video)), only(filter([action@{a_2}], video)))	explanatory	world	action	verify
how would {a} change the state of {o}?	query([change, obj@{o}, after$], only(filter([action@{a}], video)))	explanatory	world	state	query
what is the precondition of changing the {f} of {o}?	query([change, change@{f}, obj@{o}, before$], only(filter([change, change@{f}, obj@{o}], video)))	explanatory	world	state	query
is the person's action of {a_1} depending on the other person's action {a_2}?	depend(only(filter([action@{a_2}], filter([others], video))), only(filter([action@{a_1}], video)))	explanatory	multi-agent	action	verify

Table 5: Indirect references to objects and templates.

Type	Text	Indirect Reference Program
action	the action after he/she {a}	⟨query([action], iterate_until(forward, localize(after, only(filter([action@{a}], video)), filter([], video))))⟩
	the action before he/she {a}	⟨query([action], iterate_until(backward, localize(before, only(filter([action@{a}], video)), filter([], video))))⟩
	the first action in the intervals	⟨query([action], iterate_until(forward, filter([], video)))⟩
	the last action in the intervals	⟨query([action], iterate_until(backward, filter([], video)))⟩
object	the object that has status change	⟨query([change, obj$], only(filter([change, obj$], video)))⟩
	the first object that has status change	⟨query([change, obj$], iterate_until(forward, filter([change, obj$], video)))⟩
	the last object that has status change	⟨query([change, obj$], iterate_until(backward, filter([change, obj$], video)))⟩

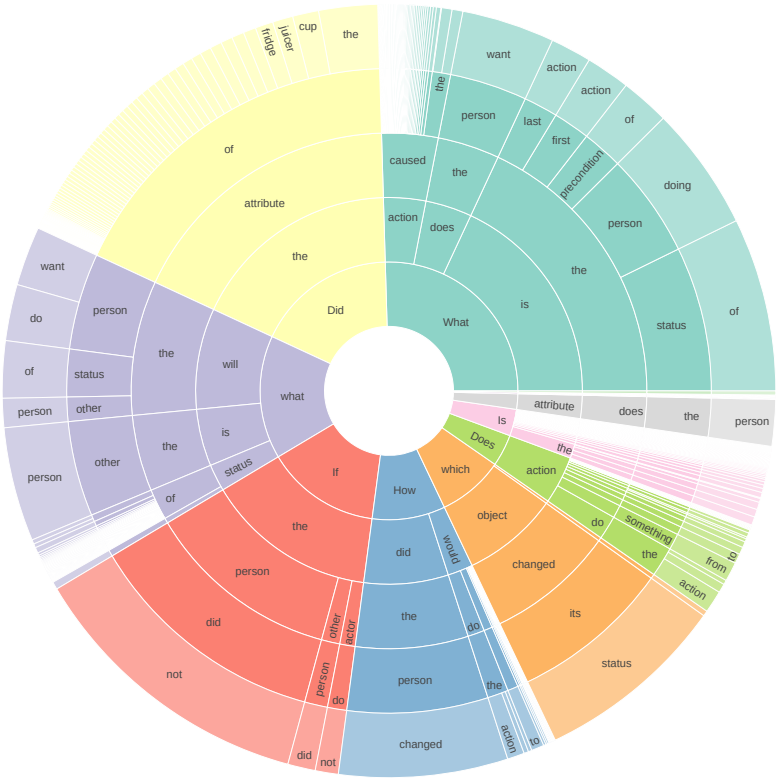

Figure 4: Questions in EgoTaskQA targets different aspects of goal-oriented reasoning with various questions.

- **HGA**: We follow the model setting provided by [7]. More specifically, We sample 20 frames per video uniformly and extract appearance features with VGG. For motion features, we sample 8 clips from each video and sample 16 frames from each clip to extract motion features with C3D [8]. We embed the text with Glove [9] embeddings to generate text tokens. We adopt the experimental configuration designed for Frame-QA on T-GIF [10] and set the learning rate to 1×10^{-4}. We train the model for 100 epochs with a batch size of 64.
- **HME**: We follow the model setting provided by [11]. As done similarly in HGA, we preprocess visual inputs with VGG and C3D for appearance and motion features and embed textual inputs with Glove embeddings. We adopt the experimental configuration designed for MSRVTT-QA [12]. We set the learning rate to 1×10^{-3} and train the model for 25 epochs with a batch size of 32.
- **PSAC**: We follow the model setting provided by [13]. We uniformly sample 20 frames per video and extract appearance features using ResNet for visual input and embed textual inputs with Glove embeddings. We adopt the experimental configuration designed for Frame-QA on T-GIF. We set the learning rate to 1×10^{-3} and train the model for 50 epochs with a batch size of 32.
- **HCRN**: We follow the model setting provided by [14]. We divide every video into eight equal length clips. Each clip is consisted of 16 frames and is used for obtaining two sources of information: frame-wise appearance feature extracted by ResNet, and motion feature extracted by ResNeXt. Textual inputs are embedded with Glove embeddings. We adopt the experimental configuration designed for MSRVTT-QA [12]. We set the learning rate to 1×10^{-4} and train the model for 50 epochs with a batch size of 32.
- **ClipBERT**: We follow the model setting provided by [15]. We preprocess the videos and texts into the ClipBERT format and adopt the experimental configuration designed for MSRVTT-QA [12]. As we have found instabilities during training (*i.e.* NaNs and Infs in gradients), we reduce the

learning rate for both transformers and CNN to 2×10^{-5}. We train the ClipBERT model for 25 epochs with a batch size of 24.

To provide a clear picture of all experimental settings and hyperparameters selected, we list training information for each model in Tab. 6. We run all experiments on a single NVIDIA A100 (80G) GPU. We provide all codes, checkpoints, and instructions for reproducing the experiments on our website.

Table 6: Hyper-parameters for baseline models evaluated on EgoTaskQA.

Model	Visual Input	Textual Input	Batch Size	Learning Rate	Training Iterations
BERT	None	BERT embedding	32	5×10^{-5}	25 epochs
VisualBERT	ResNet	BERT embedding	32	5×10^{-5}	40 epochs
HGA	VGG+C3D	Glove embedding	64	1×10^{-4}	100 epochs
HME	VGG+C3D	Glove embedding	32	1×10^{-3}	25 epochs
PSAC	ResNet	Glove embedding	32	1×10^{-3}	50 epochs
HCRN	ResNet + ResNext	Glove embedding	32	1×10^{-4}	50 epochs
HCRN w/o vision	None	Glove embedding	32	1×10^{-4}	40 epochs
ClipBERT	Grid Feature ResNet [16]	ClipBERT pretrained embedding	24	2×10^{-5}	25 epochs

D Data Documentation

We follow the datasheet proposed in [17] for documenting our EgoTaskQA benchmark:

1. Motivation
   (a) For what purpose was the dataset created?
   This dataset was created to study goal-oriented task understanding in egocentric videos. Previous works lacks the data task-related object state and relationship annotations, as well as a good evaluation metric for such information.
   (b) Who created the dataset and on behalf of which entity?
   This dataset was created by Baoxiong Jia, Ting Lei, Song-Chun Zhu and Siyuan Huang. At the time of creation, Baoxiong was a Ph.D. student at the University of California, Los Angeles (UCLA), Ting was an undergraduate student at Peking University (PKU), Siyuan was a research scientist at Beijing Institute of General Artificial Intelligence (BIGAI), and Song-Chun was a professor at UCLA, PKU, TsingHua University and BIGAI.
   (c) Who funded the creation of the dataset?
   The creation of this dataset was funded by BIGAI.
   (d) Any other Comments?
   A: None.
2. Composition
   (a) What do the instances that comprise the dataset represent?
   For video data, each instance is a video clip provided in previous work LEMMA [1]. These videos record daily indoor activities. For question-answer pairs, each instance is consist of the question text, corresponding video interval, question scope, question type, targeting answer semantic and the program.
   (b) How many instances are there in total?
   We crop videos in LEMMA into 2K video intervals for question answering. There are 40K question-answer pairs in total.
   (c) Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?
   We filter videos from the LEMMA dataset by removing videos with erroneous action annotations.
   (d) What data does each instance consist of?
   See 2.(a).
   (e) Is there a label or target associated with each instance?
   See 2.(a)
   (f) Is any information missing from individual instances?
   No.

(g) Are relationships between individual instances made explicit?
Video clips are related in the tasks performed in each videos as well as the performers. Question-answer pairs are related according to their metadata.

(h) Are there recommended data splits?
For question answering, we provide two data splits *normal* and *indirect*. Refer to Sec. 3.2 for more details.

(i) Are there any errors, sources of noise, or redundancies in the dataset?
There are almost certainly some errors in video anntotations and question-answer pairs. We did our best to minimize these, but some certainly remain.

(j) Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?
The dataset is self-contained.

(k) Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)?
No.

(l) Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?
No.

(m) Does the dataset relate to people?
Yes, all videos are recordings on human activities and all questions are related to these activities.

(n) Does the dataset identify any subpopulations (e.g., by age, gender)?
No.

(o) Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?
Yes, we can recognize the actors in the original LEMMA recordings.

(p) Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?
No.

(q) Any other comments?
None.

3. Collection Process

(a) How was the data associated with each instance acquired?
We use the videos in LEMMA and generate question-answer pairs programmatically.

(b) What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?
We use Amazon Mechanical Turk (AMT) to augment the original annotations in LEMMA.

(c) If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?
See 2.(c).

(d) Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?
For video annotations, workers are paid at a rate of 1$ per action annotation.

(e) Over what timeframe was the data collected?
The videos were recorded by LEMMA and the question answer pairs were generated in summer 2022.

(f) Were any ethical review processes conducted (e.g., by an institutional review board)?
No review processes were conducted with respect to the collection and annotation of this data.

(g) Does the dataset relate to people?
Yes, see 2.(m).

(h) Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?
We build websites for AMT workers to annotate the videos.

(i) Were the individuals in question notified about the data collection?
Yes, we instruct the AMT workers to annotate all time-varying objects in the video intervals, as well as all multi-agent relationships on visibility and awareness.

(j) Did the individuals in question consent to the collection and use of their data?
Yes, they were paid for these video annotations.

(k) If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?
Yes, this is guaranteed by AMT.

(l) Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?
No, all annotations are on objective world states with no subjective opinion or arguments involved.

(m) Any other comments?
None.

4. Preprocessing, Cleaning and Labeling

(a) Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?
No, we annotated directly on the videos. For question-answer pair generation, we operate directly on the annotated videos.

(b) Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?
Yes, we provide the raw data on our website.

(c) Is the software used to preprocess/clean/label the instances available?
For video annotation, we adopt templates from AMT. We provide all other codes on our website.

(d) Any other comments?
None.

5. Uses

(a) Has the dataset been used for any tasks already?
No, the dataset is newly proposed by us.

(b) Is there a repository that links to any or all papers or systems that use the dataset?
Yes, we provide the link to all related information on our website.

(c) What (other) tasks could the dataset be used for?
The annotated videos could also be used for world model learning. The generated question-answer pairs could also be used for evaluating models' compositional reasoning capabilities.

(d) Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?
We propose to annotate the before/after status of each object given a video. We believe this could serve as a general protocol for annotating changing world states.

(e) Are there tasks for which the dataset should not be used?
The usage of this dataset should be limited to the scope of activity or task understanding with its various downstream tasks (*e.g.* action recognition, anticipation, state/relationship recognition and question answering).

(f) Any other comments?
None.

6. Distribution

(a) Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?
Yes, the dataset will be made publicly available.

(b) How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?
The dataset could be accessed on our website.

(c) When will the dataset be distributed?
The dataset will be released to the public upon acceptance of this paper. We provide private links for the review process.

(d) Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?
We release our benchmark under CC BY-NC-SA[2] license.

(e) Have any third parties imposed IP-based or other restrictions on the data associated with the instances?
No.

(f) Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?
No.

(g) Any other comments?
None.

7. Maintenance

(a) Who is supporting/hosting/maintaining the dataset?
Baoxiong Jia is maintaining.

(b) How can the owner/curator/manager of the dataset be contacted (e.g., email address)?
E-mail addresses are at the top of the paper.

(c) Is there an erratum?
Currently, no. As errors are encountered, future versions of the dataset may be released and updated on our website.

(d) Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances')?
Yes, see 7.(c).

(e) If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?
No.

(f) Will older versions of the dataset continue to be supported/hosted/maintained?
Yes, older versions of the benchmark will be maintained on our website.

(g) If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?
Yes, errors may be submitted to us through email.

(h) Any other comments?
None.

References

[1] Baoxiong Jia, Yixin Chen, Siyuan Huang, Yixin Zhu, and Song-chun Zhu. Lemma: A multi-view dataset for learning multi-agent multi-task activities. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 8

[2] Wei Liang, Yibiao Zhao, Yixin Zhu, and Song-Chun Zhu. What is where: Inferring containment relations from videos. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3418–3424, 2016. 2

[3] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[2] https://paperswithcode.com/datasets/license

[4] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

[5] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. AGQA: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 5

[6] Hugging face visualbert implementation. https://huggingface.co/docs/transformers/model_doc/visual_bert. 5

[7] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 7

[8] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 7

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 7

[10] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[11] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[12] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of ACM International Conference on Multimedia (MM)*, 2017. 7

[13] 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 *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 7

[14] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[15] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[16] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[17] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. 8