

Supplementary Materials of SOK-Bench: A Situated Video Reasoning Benchmark with Aligned Open-World Knowledge

1. Graph Generation Algorithms

The generation processes are shown in Algorithm 1 for **Situated Knowledge Graph (SKG)**, Algorithm 2 for **General Knowledge Graph (GKG)**, and Algorithm 3 for **Situated Commonsense Knowledge Graph (SCKG)**. Please find the algorithms in the last page in details.

2. Question Templates for Bottom-up Generation

Table 1 provides an overview of the question template types. There are six primary types: Object counterfactual, Object contribution, Object reasoning, Action counterfactual, Action purpose, and Action possibility. Within each main type, we further develop subtypes by incorporating spatiotemporal reasoning, such as considering object attributes and object-object relationships (e.g., “+ obj attribute + obj-obj relationship”). Our benchmark is primarily constructed through the generation of question-answer pairs using template-based bottom-up approaches.

3. Case Studies for Top-down Generation

By utilizing a prompt template of the form $\langle \text{video content, integrated graph, QA examples, generation goal, output format} \rangle$, we effectively guide the LLM in generating multi-hop questions based on multiple edges extracted from the *integrated graph*. This prompt design, as depicted in Table 2, offers several advantages. Firstly, it enables the generation of diverse and high-quality questions and corresponding answers. Secondly, it facilitates the alignment between the generated QA pairs and the underlying knowledge graphs by instructing the LLM to demonstrate the alignment of edges in the graph (e.g., $\langle 1 \rangle$, $\langle 23 \rangle$) with the QA pairs, thereby enhancing the interpretability and reasoning steps. Please see Figure 1 for QA examples.

In the absence of this prompt template, when LLM is directly prompted to generate questions based solely on the video content, there is a risk of obtaining simplistic questions that lack situated common-sense reasoning (e.g., “What ingredients did the person use to cook mapo

Table 1. Question template types for Bottom-up QA generation

| | |
|--|---|
| Object counterfactual (e.g., What would happen if the person did not use...) | vanilla (obj) + obj attributes + obj-obj relationships + obj attribute + obj-obj relationship + obj general knowledge |
| Object contribution (e.g., Why the person use...) | vanilla (obj) + obj attributes + obj-obj relationships + obj attribute + obj-obj relationship + obj general knowledge |
| Object inference (e.g., What are the two objects which <interaction>) | |
| Action counterfactual (e.g., What would happen if the person did not...) | vanilla (act) + before / after action + before / after action + obj attr / rela |
| Action purpose (e.g., Why the person...) | vanilla (act) + before/after action + before / after action + obj attr / rela |
| Action possibility (e.g., What mistake might the person make if <possibility>) | |

tofu?”) or questions that cannot be answered based on the video (e.g., “Who will eat the mapo tofu?”). While our approach does not entirely eliminate such situations, our prompt design significantly mitigates this issue. Although there may still be instances where the LLM generates relatively straightforward questions (e.g., “What is the color of tofu?”), our method ensures that the generated questions are correct, as we constrain the LLM’s output within the boundaries defined by the knowledge graphs.

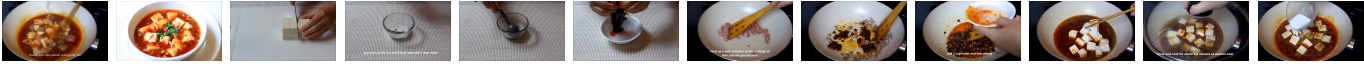
4. Object and Action Overview

Our benchmark includes 1005 objects and 505 actions. The word clouds of objects and actions are presented in Figure 2 and Figure 3 respectively.

5. Data Quality Evaluation and Validation

We enlisted the assistance of human evaluators to assess the quality of our benchmark through visual inspection.

Example QAs by top-down generation



Question: Why is transparent substance important in the recipe for mapo tofu?

A. It is made up of two hydrogen atoms and one oxygen atom.
B. It helps to create the sauce, hydrate and soften tofu, and enhance flavor absorption.✓
 C. It is important only because it dissolves and distributes seasonings.
 D. It is important in the recipe for mapo tofu because it is essential for human survival.

Question: How would the taste of the dish change if you skip adding ground pork and salt?

A. The dish would lack the meatiness and seasoning, resulting in a less flavorful mapo tofu.✓
 B. The dish would taste even spicier.
 C. The dish would become more authentic mapo tofu.
 D. The dish would taste the same.

Question: How would the dish change if the ingredient fried with mapo tofu paste was replaced with chicken?

A. The dish would taste exactly the same.
 B. The dish would be no more mapo tofu.
 C. The dish would be more flavorful.
D. Chicken would alter the traditional taste of mapo tofu.✓

Figure 1. Example of top-down generated question-answer pairs.

Table 2. Exemplary prompts for top-down QA generation

| | |
|------------------|--|
| video content | #System#: You are a helpful, pattern-following assistant. You are very good at deciphering a video, such as summarizing video content, figuring out related commonsense knowledge, and identifying the inner logic. As a commonsense-driven assistant, your task is to analyze a video (length: 119 seconds) that showcases a person introducing 6 cooking steps of mapo tofu. The steps are described in a specific format of '[Step N (from <timestamp> to <timestamp>): detailed description of cooking actions]' as follows: ... |
| integrated graph | #System#: From the <video>, use your commonsense reasoning ability, you can get a <knowledge graph> shown in JSON format: <pre>{<0>: {'head': 'tofu', 'edge': 'counterfactual', 'tail': 'The dish would no longer be mapo tofu.'}, <1>: {'head': 'tofu', 'edge': 'contribution', 'tail': ' adds texture and absorbs the flavors of the sauce in mapo .'}, ... <6>: {'head': 'cut tofu into square pieces', 'edge': 'things related to', 'tail': 'tofu' } ... }</pre> |
| QA examples | #User#: Based on video content and <knowledge graph>, use your creativity, generate a HARD question which involves reasoning involving multiple indexes from <knowledge graph>. The description of the question should be natural. Then, answer the question with correct answer, provide 3 wrong answers. Output in JSON format with key 'question', 'correct answer', '3 wrong answers'. Others cannot answer the question unless they have known <video content>. #Assistant#: {question: <Ingredient A>is mixed with <Ingredient B>to create a slurry, which is added to the to thicken the sauce. What are <Ingredient A>and <Ingredient B>?, correct answer: cornstarch and water, 3 wrong answers: ...} ... |
| generation goal | #User#: Based on video content and <knowledge graph>, use your creativity, generate a HARD question which involves reasoning involving multiple indexes from <knowledge graph>. |
| output format | #User#: The description of the question should be natural. Then, answer the question with correct answer, provide 3 wrong answers. Output in JSON format with key 'question', 'correct answer', '3 wrong answers'. Others cannot answer the question unless they have known <video content>. |

From each of the 12 question types, we validated all 44K question-answer pairs with the effort from 8 non-CS PhD students with STEM skills, degrees and responsible attitudes, ensuring no conflict of interest. The quality validation process consisted of two stages. In the first stage, the evaluators examined whether the questions and answers were correct based on its comprehension, semantics, and logic. This entailed verifying the accuracy of the answer, determining if the question was answerable, and assessing its relevance to the accompanying video (refer to the “Correctness” column in Table 3). The 3-voter validation was carried out. Overall, human evaluation indicated

92.05%±0.2% of the generated pairs as correct with a 99% confidence level. In the second stage, for all the “correct” question-answer pairs, the evaluators further assessed whether the question could be answered without watching the video. For instance, consider the following example: “Question: What is the color of tofu? Answer: White.” Although this question-answer pair is correct and related to the video on cooking mapo tofu, the question can be answered solely through commonsense and does not require watching the video. The evaluators calculated the proportion of questions that could only be answered by watching the video (refer to the “Need video to answer” column in Table 3).

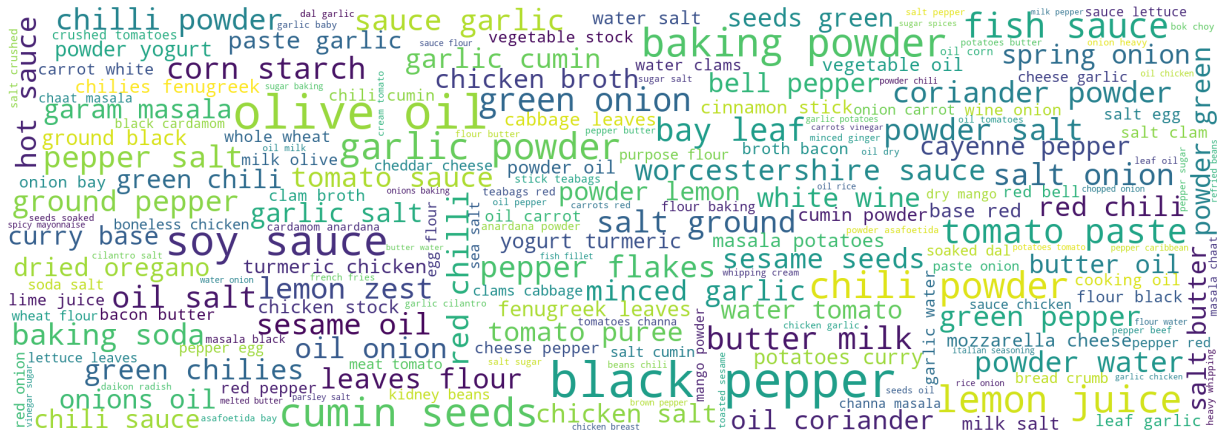


Figure 2. Word cloud of objects presented in our benchmark.



Figure 3. Word cloud of actions presented in our benchmark.

The results indicate that our benchmark exhibits a high level of correctness and a high ratio of questions that necessitate watching the video for answering. A high correctness ratio signifies the validity of our automated question-answer generation approach. A high ratio of questions requiring video comprehension demonstrates that the majority of generated questions demand both spatiotemporal reasoning based on the video content and situated common-sense reasoning. This accomplishment aligns with our objective in designing the benchmark.

6. Experiment Settings

For deployable models (Video-LLaMa, Video-ChatGPT, PandaGPT and AskAnything), we use their pre-trained 7B versions, respectively. For GPT4-Vision, we provide 5 frames chosen equidistantly in each video.

When evaluating these models in multiple choice setting, we observe that sometimes they do not give explicit choices, but instead answers the question directly (see examples). Therefore, we map an answer to a choice by:

- If the answer can be parsed into an integer choice successfully, use it as the model’s choice.

Table 3. QA quality validation

| QA type | Correctness ¹ | Need video to answer ² |
|---|--------------------------|-----------------------------------|
| object counterfactual + spatiotemporal | 0.91 | 0.98 |
| object contribution + spatiotemporal | 0.92 | 0.98 |
| object counterfactual + general knowledge | 0.95 | 0.88 |
| object contribution + general knowledge | 0.94 | 0.89 |
| action purpose + spatiotemporal | 0.92 | 0.98 |
| action counterfactual + spatiotemporal | 0.88 | 0.99 |
| object inference | 0.95 | 0.95 |
| object counterfactual | 0.92 | 0.94 |
| action counterfactual | 0.91 | 0.95 |
| action purpose | 0.95 | 0.96 |
| action possibility | 0.98 | 0.96 |
| object contribution | 0.94 | 0.92 |

¹ Correctness = num of correct QA pairs / total num of QA pairs

² Need video to answer = num of QA pairs which can only be correctly answered by watching the video / total num of QA pairs

- Otherwise, we compute the BLEU score between each choice and the model’s output, and choose the maximum as its choice.
- We notice that API models generally do not need this fix, as they show better instruction following abilities.

Please refer to Figure 4 for an illustration of example

results obtained by the baseline models on our benchmark.

7. Ablation Studies

7.1. Effectiveness of Prompting Generation.

By referring to Table 3, it becomes evident that when incorporating spatiotemporal information (*i.e.*, “+spatiotemporal”), nearly all the questions can only be answered by watching the accompanying videos. This finding effectively validates the significance of MLLMs in generating top-notch situated commonsense questions.

As indicated in Table 4, when prompts include a specified knowledge *aspect*, a broader range of knowledge is generated. By incorporating *general knowledge examples* (π) and *output format* (ϕ), we are able to generate the same number of knowledge pieces using significantly fewer tokens, while ensuring that all the generated content can be parsed in JSON format (see Table 5).

$\{\pi_i^{obj}\}$, $\{\pi_i^{act}\}$, generation goal instructions $\{\omega\}$, output format instructions $\{\phi\}$

Table 4. Number of pieces of generated knowledge per object with or without knowledge aspect.

| | Num of knowledge / obj |
|------------|------------------------|
| w/o aspect | 8 |
| w/ aspect | 26 |

Table 5. Number of tokens used and ratio of JSON parsable content by different prompt strategies

| | Num of tokens / vid | JSON parse ¹ |
|---------------------|---------------------|-------------------------|
| Gen Goal | 7090 | 0% |
| + Format | 4559 | 11.1% |
| + Examples + Format | 3512 | 100% |

¹ The ratio represents the number of outputs that can be parsed as JSON format divided by the total number of outputs

7.2. Effectiveness of Few-Shot Self-Prompting.

We present a comparison between two scenarios: one involving Few-Shot Self-Prompting and the other without, as depicted in Table 6. The table clearly demonstrates that the utilization of Few-Shot Self-Prompting techniques enables LLM to generate significantly more specific and concrete situated commonsense knowledge.

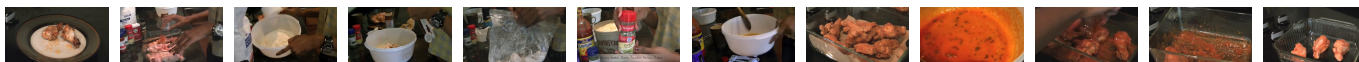
Table 6. Comparison between two scenarios: one with Few-Shot Self-Prompting and one without. A case about cooking buffalo wings is shown here. Column of “object contribution to cooking” shows the generated contents by LLM.

| | objects | object contribution to cooking |
|-----------------------|--------------------|---|
| w/o FSSP ¹ | wings | are rinsed, seasoned, cooked, and mixed with buffalo sauce in the cooking process |
| | garlic powder | adds flavor to the buffalo wings |
| | chili powder | adds flavor to the buffalo wings |
| | paprika | adds flavor and a hint of smokiness to the buffalo wings |
| | salt | enhances the flavor of the buffalo wings |
| | cayenne pepper | adds spiciness to the buffalo wings |
| | buffalo wing sauce | adds flavor to the wings |
| w/ FSSP ² | wings | are the main ingredient and are cooked until crispy before being coated in buffalo sauce |
| | garlic powder | is used as a seasoning for the wings, adding a savory and aromatic flavor to the dish |
| | chili powder | adds a spicy kick to the wings, enhancing the overall flavor and heat of the dish |
| | paprika | adds a smoky and slightly sweet flavor to the wings, enhancing their overall taste and appearance |
| | salt | enhances the flavor of the wings, bringing out the natural taste of the meat and other seasonings |
| | cayenne pepper | adds a spicy and fiery flavor to the wings, giving them a delicious and bold taste |
| | buffalo wing sauce | adds a tangy and zesty flavor to the wings, giving them their signature taste and texture |

¹ w/o FSSP: without Few-Shot Self-Prompting

² w/ FSSP: with Few-Shot Self-Prompting

A video showing the cooking process of Buffalo Wings



(a) Model: GPT-4 Vision

Question: Look What would happen if the person did not use the thing which also can be used to widely used in hungarian, spanish, and indian cuisines in culinary traditions and central and south america in origin during the cooking process?

GT answer: The buffalo wings would lack the smoky and slightly sweet flavor that paprika adds.

Model output: If the person did not use salt, which is commonly used in Hungarian, Spanish, Indian cuisines, and originates from Central and South America, the dish would likely have less flavor and taste bland, as salt is a basic seasoning that enhances the taste of food.

(b) Model: Valley

Question: What would happen if the person did not use the thing which also can be used to varies (odorless to distinct) in odor and varies (depending on the type of oil) in boiling point during the cooking process? Please choose from the following choices (0 or 1 or 2 or 3). 0: The Thai fried rice would lack the necessary lubrication and frying element, resulting in a dry and less flavorful dish; 1: The Thai fried rice would lack the distinctive and aromatic flavors that spices provide, resulting in a less flavorful dish; 2: The Thai fried rice would lack a significant source of protein and the overall flavor profile would be affected; 3: The Thai fried rice would lack texture, color, and essential nutrients, resulting in a less balanced and flavorful dish. Please give only a number chosen from 0, 1, 2, or 3.

GT answer: 0: The Thai fried rice would lack the necessary lubrication and frying element, resulting in a dry and less flavorful dish.

Model output: Based on the video, if the person did not use the spoon to mix the ingredients in the frying pan, the dish would not be well-combined and the flavors might not be evenly distributed. The spoon is an essential tool in the cooking process, as it helps to ensure that the ingredients are mixed properly and cooked evenly. Without the spoon, the dish would likely be less appetizing and less enjoyable to eat.

(c) Lateral comparison

Question: What would happen if the person did not use the thing which also can be used to widely used in hungarian, spanish, and indian cuisines in culinary traditions and central and south america in origin during the cooking process? Please choose from the following choices (0 or 1 or 2 or 3). 0: The buffalo wings would be less spicy and have a milder flavor without black pepper; 1: The flavor of the buffalo wings would be slightly different without the garlic powder; 2: The buffalo wings would lack the smoky and slightly sweet flavor that paprika adds; 3: The buffalo wings would not have the same crispy texture without the oil. Please give only a number chosen from 0, 1, 2, or 3.

GT answer: 2: The buffalo wings would lack the smoky and slightly sweet flavor that paprika adds

#GPT4-V#: 2.

#Ask Anything#: 2.

#PandaGPT#: If the person did not use the sauce that can be widely used in Hungarian, Spanish, and Indian cuisines in culinary traditions and Central and South America in origin, the buffalo wings would have a less spicy and milder flavor without black pepper (0). The flavor of the buffalo wings would be slightly different without garlic powder (1), but still have the smoky and slightly sweet flavor that paprika adds (2). The buffalo wings would lack the crispy texture without the oil (3).

#Valley#: 1: The person is using a spoon to stir the food in the pot.

#VideoLLaMa#: (corrupted output)

Figure 4. (a): An example showcasing the GPT-4 Vision results in a direct-answer setting. (b): A typical instance where the model's output lacks a specific choice but provides an explanation in a multichoice setting. To determine the correctness of the model's output, we compute the similarity between the model's output and each available option. (c): A comparative analysis conducted among the baseline models..

Algorithm 1: Situated Knowledge Graph

Input: video v , manual annotations of events $\{y_t\}$, Large Language Model LLM, Vision-Language Model VL, generation goal instructions $\{\omega\}$, output format instructions $\{\phi\}$

1 **Function** GetSKG($v, \{y_t\}$):

Init: Situated Knowledge Graph $\mathbf{G}_v = []$

 /* 1 parse objs and acts from annotations and output in JSON format */

2 **objs** $\{o_i\} \leftarrow \text{LLM}(\langle \{y_t\} \mid \omega_{obj}, \phi_{obj} \rangle)$;

3 **acts** $\{a_t\} \leftarrow \text{LLM}(\langle \{y_t\} \mid \omega_{act}, \phi_{act} \rangle)$;

 /* 2 obtain obj attributes and obj-obj relationships */

4 **sample frames** with timestamps $\{f_{t^*}\} \leftarrow v$;

5 **for each** $f_{t^*} \in \{f_{t^*}\}$ **do**

6 extract **obj** strings from frames

7 $o^* \leftarrow \text{VL}(\langle f_{t^*} \mid \omega_{obj}^* \rangle)$;

8 **parse objs** in JSON format

9 $\{o_i^*\} \leftarrow \text{LLM}(\langle o^* \mid \phi_{obj}^* \rangle)$;

 // check missing objs (VL may fail to recognize objs in frames)

10 **for each** $o_i \in \{o_i\}$ **do**

11 **if** o_i in t^* & $o_i \notin \{o_i^*\}$ **then**

12 $\{o_i^*\} := \{o_i^*\} + o_i$

 // obj attributes

13 **for each** $o_i^* \in \{o_i^*\}$ **do**

14 **get obj attributes**

15 $attr_i \leftarrow \text{VL}(\langle f_{t^*} \mid \omega_{attr}^i \rangle)$;

 // add edges of <time, obj> and <obj, attribute> to SKG

16 add edge (t^*, o_i^*) to \mathbf{G}_v ;

17 add edge ($o_i^*, attr_i$) to \mathbf{G}_v ;

 // obj-obj spatial relations

18 **for each pair** (o_i^*, o_j^*) **do**

19 **get obj-obj spatial relations**

20 $re_{i,j} \leftarrow \text{VL}(\langle f_{t^*}, \omega_{re}^{i,j} \rangle)$;

 // add edge of <obj i, relationship, obj j> to SKG

21 add edge ($o_i^*, re_{i,j}, o_j^*$) to \mathbf{G}_v ;

 /* 3 align acts with objs */

22 **for each** $f_{t^*} \in \{f_{t^*}\}$, **each** $a_t \in \{a_t\}$ **do**

23 **for each** $o_i^* \in \{o_i^*\}$ **do**

24 **if** t^* in t & o_i^* in a_t **then**

25 // add edges of <act, obj> to SKG

26 add edge (a_t, o_i^*) to \mathbf{G}_v ;

 // add edges of <time, act> to SKG

26 add edge (t, a_t) to \mathbf{G}_v ;

27 **return** \mathbf{G}_v ;

Algorithm 2: General Knowledge Graph

Input: Situated Knowledge Graph \mathbf{G}_v , Large Language Model LLM, general knowledge aspects $\{\eta_i^{obj}\}$, $\{\eta_i^{act}\}$ with corresponding knowledge examples $\{\pi_i^{obj}\}$, $\{\pi_i^{act}\}$, generation goal instructions $\{\omega\}$, output format instructions $\{\phi\}$

1 **Function** GetGKG(\mathbf{G}_v):

Init: General Knowledge Graph $\mathbf{G}_{gk} = []$

2 extract **objs** $\{o_i\}$, **acts** $\{a_t\}$ from \mathbf{G}_v ;

 /* 1 general knowledge of objs */

3 **for** $o_i \in \{o_i\}$ **do**

 // e.g., knowledge aspects of physical, chemical, usage, ...

4 **for** $\eta_j^{obj} \in \{\eta_j^{obj}\}$, $\pi_j^{obj} \in \{\pi_j^{obj}\}$ **do**

5 construct generation goal based on one aspect

6 $\omega_{obj}^* \leftarrow \langle o_i \mid \omega_{obj}, \eta_j^{obj} \rangle$;

7 generate **knowledge** in desired JSON format

8 $\kappa_i^j \leftarrow \text{LLM}(\langle \pi_j^{obj}, \omega_{obj}^*, \phi_{obj} \rangle)$;

9 add edge ($o_i, \eta_j^{obj}, \kappa_i^j$) to \mathbf{G}_{gk} ;

 /* 2 general knowledge of acts */

10 **for** $a_t \in \{a_t\}$ **do**

11 **for** $\eta_j^{act} \in \{\eta_j^{act}\}$, $\pi_j^{act} \in \{\pi_j^{act}\}$ **do**

12 construct generation goal based on one aspect

13 $\omega_{act}^* \leftarrow \langle a_t \mid \omega_{act}, \eta_j^{act} \rangle$;

14 generate **knowledge** in desired JSON format

15 $\kappa_i^j \leftarrow \text{LLM}(\langle \pi_j^{act}, \omega_{act}^*, \phi_{act} \rangle)$;

16 add edge ($a_t, \eta_j^{act}, \kappa_i^j$) to \mathbf{G}_{gk} ;

17 **return** \mathbf{G}_{gk} ;

Algorithm 3: Situated Commonsense Knowledge Graph

Input: Situated Knowledge Graph \mathbf{G}_v , General Knowledge Graph \mathbf{G}_{gk} , Large Language Model LLM, situated commonsense knowledge aspects $\{\eta_i^{obj}\}$, $\{\eta_i^{act}\}$, generation goal instructions $\{\omega\}$, output format instructions $\{\phi\}$

1 **Function** GetSCKG (\mathbf{G}_v , \mathbf{G}_{gk}):

Init: Situated Commonsense Knowledge Graph
 $\mathbf{G}_{sk} = [], J = 2$

2 extract **objs** $\{o_i\}$, **acts** $\{a_t\}$ from \mathbf{G}_v ;

3 $g_v \leftarrow \text{texturalized } \mathbf{G}_v$;

 /* 1 sc knowledge of objs */

4 **for each** $\eta_j^{obj} \in \{\eta_j^{obj}\}$ **do**

 // e.g., knowledge aspects of counterfactual and obj contribution

 situated commonsense examples $\pi_j = []$;

repeat

 7 **for** $o_i \in \{o_i\}$ **do**

 8 $k_g \leftarrow$ knowledge related to o_i in \mathbf{G}_{gk} ;

 9 $\omega_{obj}^* \leftarrow \langle o_i \mid \omega_{obj}, \eta_j^{obj} \rangle$;

 10 $\kappa_i^j \leftarrow \text{LLM}(\langle g_v, k_g, \pi_j, \omega_{obj}^*, \phi_{obj} \rangle)$;

 11 add or update κ_i^j in π_j ;

 12 add or update $(o_i, \eta_j^{obj}, \kappa_i^j)$ in \mathbf{G}_{sk} ;

13 **until** *Finishing J iterations* ;

 /* 2 sc knowledge of acts */

14 **for** $\eta_j^{act} \in \{\eta_j^{act}\}$ **do**

 // e.g., knowledge aspects of counterfactual and act purpose

 situated commonsense examples $\pi_j = []$;

repeat

 17 **for** $a_t \in \{a_t\}$ **do**

 18 $k_g \leftarrow$ knowledge related to a_t in \mathbf{G}_{gk} ;

 19 $\omega_{act}^* \leftarrow \langle a_t \mid \omega_{act}, \eta_j^{act} \rangle$;

 20 $\kappa_i^j \leftarrow \text{LLM}(\langle g_v, k_g, \pi_j, \omega_{act}^*, \phi_{act} \rangle)$;

 21 add or update κ_i^j in π_j ;

 22 add or update $(a_t, \eta_j^{act}, \kappa_i^j)$ in \mathbf{G}_{sk} ;

23 **until** *Finishing J iterations* ;

24 **return** \mathbf{G}_{sk} ;
