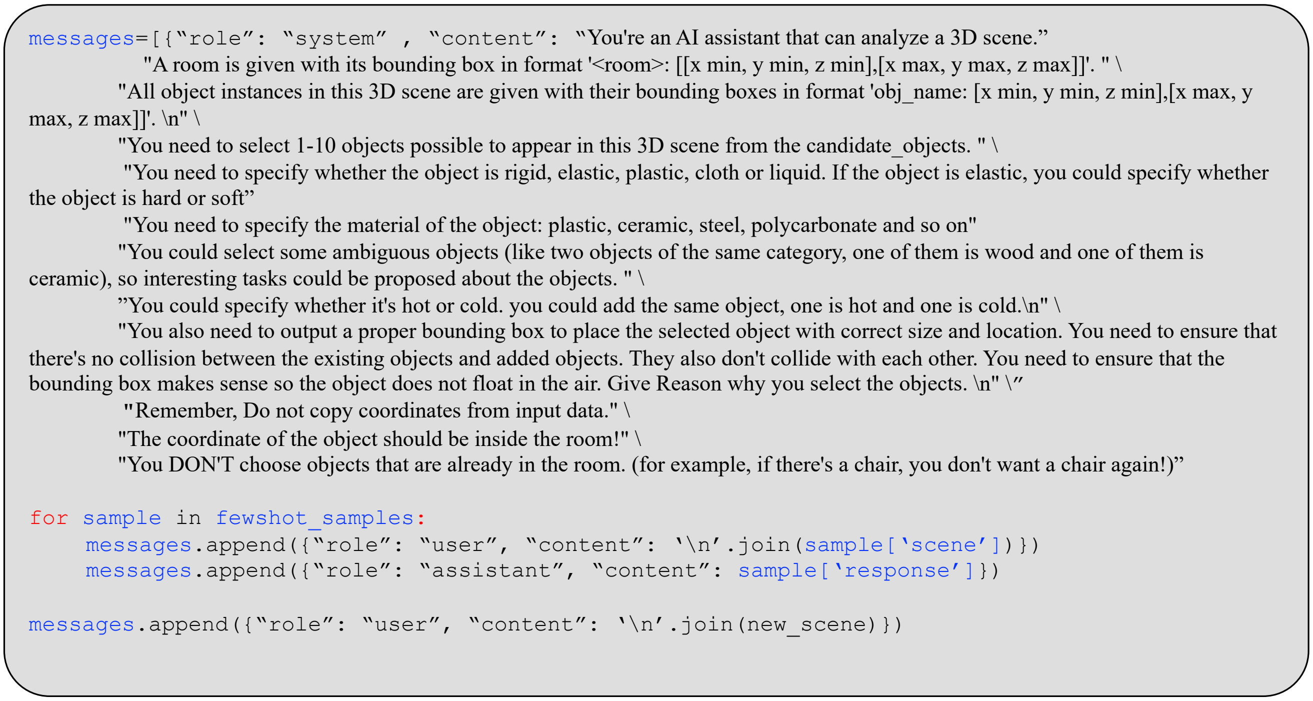
Supplementary Material for
"MultiPLY: A Multisensory Object-Centric Embodied Large Language Model in 3D World"

Contents

A. Dataset

A.1. More details on Scene Construction

In figure 1, we show how we add new objects to the HM3D scenes. Specifically, ChatGPT is asked to generate: 1) object bounding boxes; 2) object material and material properties; 3) temperatures.



```
messages=[{“role”: “system” , “content”: “You're an AI assistant that can analyze a 3D scene.”
                "A room is given with its bounding box in format '<room>: [[x min, y min, z min],[x max, y max, z max]]'. " \
            "All object instances in this 3D scene are given with their bounding boxes in format 'obj_name: [x min, y min, z min],[x max, y
max, z max]]'. \n" \
            "You need to select 1-10 objects possible to appear in this 3D scene from the candidate_objects. " \
             "You need to specify whether the object is rigid, elastic, plastic, cloth or liquid. If the object is elastic, you could specify whether
the object is hard or soft”
             "You need to specify the material of the object: plastic, ceramic, steel, polycarbonate and so on"
            "You could select some ambiguous objects (like two objects of the same category, one of them is wood and one of them is
ceramic), so interesting tasks could be proposed about the objects. " \
            ”You could specify whether it's hot or cold. you could add the same object, one is hot and one is cold.\n" \
            "You also need to output a proper bounding box to place the selected object with correct size and location. You need to ensure that
there's no collision between the existing objects and added objects. They also don't collide with each other. You need to ensure that the
bounding box makes sense so the object does not float in the air. Give Reason why you select the objects. \n" \”
             "Remember, Do not copy coordinates from input data." \
            "The coordinate of the object should be inside the room!" \
            "You DON'T choose objects that are already in the room. (for example, if there's a chair, you don't want a chair again!)”

for sample in fewshot_samples:
    messages.append({“role”: “user”, “content”: ‘\n’.join(sample[‘scene’])})
    messages.append({“role”: “assistant”, “content”: sample[‘response’]})

messages.append({“role”: “user”, “content”: ‘\n’.join(new_scene)})
```

Figure 1. Prompts for adding objects to the scene

A.2. More details on Sensor Data Acquisition

In this section, we elaborate on how we get the sensor data of the objects in details.

A.2.1 Tactile

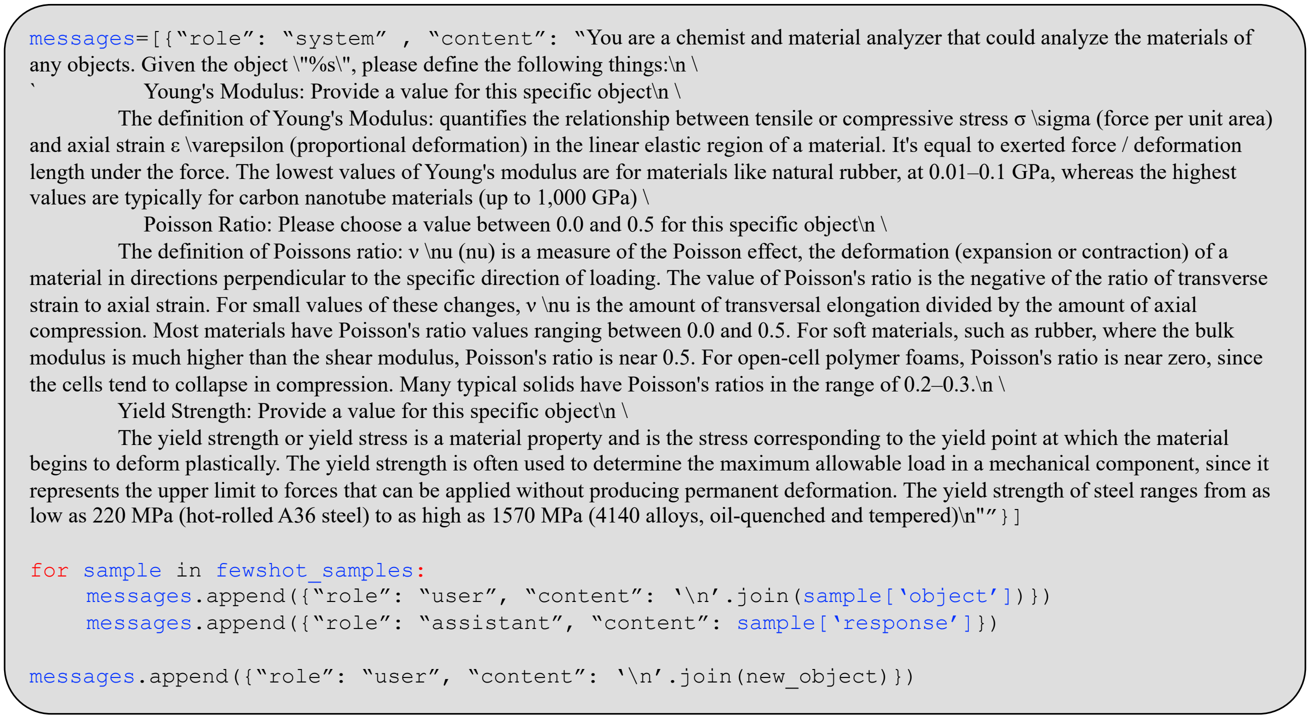
DiffTactile [1] requires us to provide a set of parameters for tactile simulation of different objects. In addition to telling to the model whether we are inputting a rigid, elastic, or elasto-plastic object, we also need to specify the parameters such as Young's modulus, Poisson's ratio, Yield Strength and so on.

As in the main paper, when ChatGPT adds objects to the scene, it also specifies what kinds of objects (*e.g*, rigid, elastic, plastic) and the softness / deformability (in the description of language) of each object. In order to get the parameters required by DiffTactile, we prompt ChatGPT with the type and the softness / deformability description, as well as detailed definition of each parameter, and the possible values of the parameters of several few-shot examples. ChatGPT is asked to return the detailed parameter combinations of the given objects. For example, a soft bread corresponds to a smaller young's modulus, while a harder one corresponds to a larger young's modulus. We add the prompt in getting the parameters in Figure 2.

We input the object into DiffTactile, normalize the shape of the gripper according to the object. We record the 2D initial position and final position of the markers in the gripper. And we turn the tactile readings into a 2D image, by drawing an arrowed line from the initial position to the final position. We show some examples of tactile images in Figure 3. We sample 16 touching positions of each object. In training and evaluation, we randomly return one image of the object.

A.2.2 Impact Sound

ObjectFolder [2] stores the multi-modal information all in implicit fields. That is, by inputting a striking location to the sound implicit field of an object, we could get the impact sound of striking the object at the specific location. For each object, we



```
messages=[{“role”: “system” , “content”: “You are a chemist and material analyzer that could analyze the materials of
any objects. Given the object \"%s\", please define the following things:\n \
`                Young's Modulus: Provide a value for this specific object\n \
            The definition of Young's Modulus: quantifies the relationship between tensile or compressive stress σ \sigma (force per unit area)
and axial strain ε \varepsilon (proportional deformation) in the linear elastic region of a material. It's equal to exerted force / deformation
length under the force. The lowest values of Young's modulus are for materials like natural rubber, at 0.01–0.1 GPa, whereas the highest
values are typically for carbon nanotube materials (up to 1,000 GPa) \
                Poisson Ratio: Please choose a value between 0.0 and 0.5 for this specific object\n \
            The definition of Poissons ratio: ν \nu (nu) is a measure of the Poisson effect, the deformation (expansion or contraction) of a
material in directions perpendicular to the specific direction of loading. The value of Poisson's ratio is the negative of the ratio of transverse
strain to axial strain. For small values of these changes, ν \nu is the amount of transversal elongation divided by the amount of axial
compression. Most materials have Poisson's ratio values ranging between 0.0 and 0.5. For soft materials, such as rubber, where the bulk
modulus is much higher than the shear modulus, Poisson's ratio is near 0.5. For open-cell polymer foams, Poisson's ratio is near zero, since
the cells tend to collapse in compression. Many typical solids have Poisson's ratios in the range of 0.2–0.3.\n \
            Yield Strength: Provide a value for this specific object\n \
            The yield strength or yield stress is a material property and is the stress corresponding to the yield point at which the material
begins to deform plastically. The yield strength is often used to determine the maximum allowable load in a mechanical component, since it
represents the upper limit to forces that can be applied without producing permanent deformation. The yield strength of steel ranges from as
low as 220 MPa (hot-rolled A36 steel) to as high as 1570 MPa (4140 alloys, oil-quenched and tempered)\n"” }]

for sample in fewshot_samples:
    messages.append({“role”: “user”, “content”: ‘\n’.join(sample[‘object’])})
    messages.append({“role”: “assistant”, “content”: sample[‘response’]})

messages.append({“role”: “user”, “content”: ‘\n’.join(new_object)})
```

Figure 2. Prompts for getting the material parameters for the objects

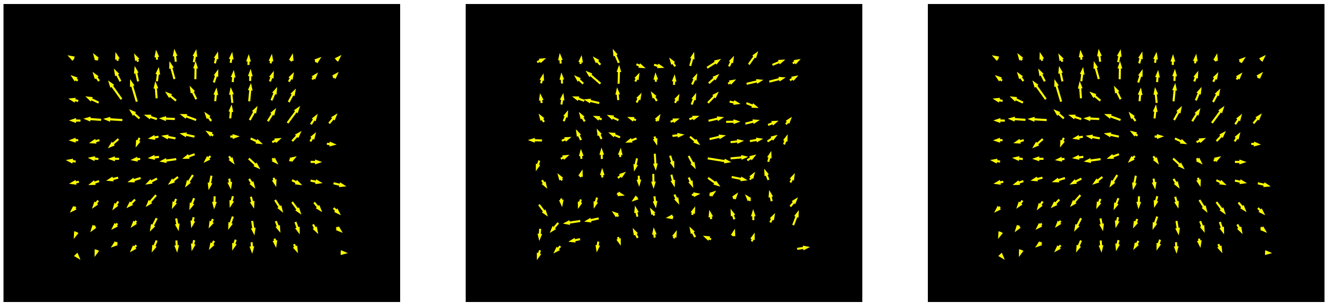


Figure 3. Examples of tactile images.

randomly sample 10 locations in the mesh points to get the impact sound. In training and evaluation, we randomly return one impact sound of the object.

A.2.3 Ambient Sound

AudioSet is paired with objects to represent ambient sound. The AudioSet ontology is organized in a hierarchy structure. From the root node to the leaf node, the description granularity becomes finer (e.g., Music - Musical instrument - Keyboard - Piano - Electric piano). Each ontology entry is attached with a description (e.g., “Glass: sounds associated with the non-crystalline amorphous solid that is often transparent and has widespread practical, technological, and decorative uses”). Each audio is labeled with multiple ontology entries tracing from the child node to the root node (e.g., the sound of the piano will be labeled with “Piano”, “Keyboard”, “Musical Instrument”, and “Music”, but without “Electric piano” since this piano is not electric). We prompt ChatGPT to match each ontology entry with object categories (Figure 4).

A.2.4 Temperature

We add the prompt in getting the temperature in Figure 5.

```
System:
You are an AI audio assistant that can analyze descriptions and tags of sounds.  The input has three fields. 'tags' are some labels about the
sound. 'description' is the text description of the sound. 'objects' are a list of candidate objects. You need to infer what kind of objects can
make the sound based on 'tags' and ‘descriptions’.  You need to select ALL objects that are possible to make this sound from the 'objects' list.
Remember, the object MUST be found in a normal INDOOR environment. Do not include objects that do not exist in the 'objects' list.
Return [] if no object satisfies the condition.

Example Questions:
tags=['Frying (food)', 'Domestic sounds, home sounds', 'Sounds of things'],
description='The sound of cooking food in oil or another fat.',
objects=[poncho, pool_table, pop_(soda), popsicle, postbox_(public), pan_(for_cooking), postcard, poster, pot, potato, potholder]

Example Answers:
[pot, pan_(for_cooking)]
```

Figure 4. Prompts to match AudioSet with Objects

```
messages=[{“role”: “system” , “content”: “You are a temperature analyzer that can analyze a 3D scene.
                You need to assign a temperature (celsius) for the input object. The default room temperature is 26 degree celsius. \
                Pay attention to the hot and cold label of the objects. For example, cup_hot can be as hot as 85 celsius, and cup_cold can be as
cold as 5 celsius.”}]

for sample in fewshot_samples:
    messages.append({“role”: “user”, “content”: ‘\n’.join(sample[‘object’])})
    messages.append({“role”: “assistant”, “content”: sample[‘response’]})

messages.append({“role”: “user”, “content”: ‘\n’.join(new_object)})
```

Figure 5. Prompts on generating temperature for each object

A.3. More details on Task Construction

In Figure 6, we illustrate the prompts for generating the language task data for Multisensory-Universe. Specifically, the actions could return the expected observation in the form of language (*e.g.,* tactile map of and object when touching). We insert that into the state tokens for placeholder, and after the agent has executed the actions in the space and gets the observations, we append the observations back to the state tokens.

Ablative Model	Acc
MultiPLY Vision	21.0
MultiPLY Audio	13.2
MultiPLY Tactile	10.5
MultiPLY Temperature	11.2
MultiPLY Vision, Audio	31.8
MultiPLY Vision, Tactile	24.3
MultiPLY Vision, Temperature	25.7
MultiPLY Audio, Tactile	20.6
MultiPLY Audio, Temperature	23.4
MultiPLY Tactile, Temperature	18.9
MultiPLY Vision, Audio, Tactile	45.3
MultiPLY Vision, Tactile, Temperature	41.4
MultiPLY Vision, Audio, Temperature	45.3
MultiPLY Audio, Tactile, Temperature	37.7
MultiPLY	56.7

Table 1. Ablative Study of MultiPLY

```
messages=[{“role”: “system” , “content”: “You are an AI assistant / task generator in the room. All object instances in
this 3D scene are given, along with their bounding boxes and ids." \
            "The bounding boxes are represented by a 3D coordinate (x, y, z) with units of meters. " \
            “If the object emits a sound, it will have a ‘emit’ label." \
            ”If the object could be hit, it will have a ’hit’ label.” \
             ”You could use the actions to interact with the environment. They are:”
               “<SELECT>: which returns the id of the object”
               “<NAVIGATE>: which navigates to the object selected”
               “<OBSERVE>: which returns the visual details of the object”
               “<TOUCH>: which returns tactile and temperature information of the object”
               ”<HIT>: which returns the impact sound of the object”
               ”<PICK UP>: pick up the object”
               “<PUT DOWN>: put down the object”
               “<LOOK AROUND>: retrieves objects ids and categories near the object”

Using the provided object instance information and selected objects, you need to generate a task that could be performed in the scene.
Exempler tasks include captioning, question answering, dialogue, manipulation, task decomposition, rearrangement. For example:
               ”Captioning: you need to choose one object, describing its information of all modalities, and also its relationships to the other
objects. ”
               “Question Answering: you need to generate several question-answering pairs about the 3D scene. The questions must be
answered by exploring the room using the above actions.”
               “Manipulation: You need to generate some manipulation tasks which you use the actions to manipulate the objects”
               “Task Decomposition: You need to design a task that could be performed in this room and decompose it into 3-10 sub-tasks.
The task must be completed using the actions.”
               “Rearrangement: If the objects are in a weird position, move them to a suitable location using the actions.”
You also need to decompose the description process by several actions to interact with the environment using the tokens above. You need
to also specify what’s the observation / feedback you could get by executing the action. For example <SELECT> -> returns apple(65),
where 65 is the object id, or touch -> returns tactile map and temperature of apple(65). You need to output your reasoning processes like “I
need to touch it ” or conclusions like “it’s hot”
for sample in fewshot_samples:
    messages.append({“role”: “user”, “content”: ‘\n’.join(sample[‘scene’])})
    messages.append({“role”: “assistant”, “content”: sample[‘response’]})

messages.append({“role”: “user”, “content”: ‘\n’.join(new_scene)})
```

Figure 6. Prompts for task construction

B. Experiments

B.1. Experimental Details

We tune the model based on the llava-v1.5-7b checkpoint of the LLaVA model. We use Adam optimizer with learning rate of 1e-6. We train the model on 4*132 V100s. We use a batch size of 2112. The training of multi-modal adapters takes 2 hours, while the whole finetuning takes less day 1 day to complete.

We use the mm projector of the original LLaVA for adapting scene representations and object point clouds to the LLM. The sound, tactile and temperature adapters are all one linear layer with input size 1024 and output size 1024.

We use the default CLIP vision encoder of LLaVA to encode all objects, point clouds, tactile and temperature images. Specifically, for objects, we use segment anything [4] to get the objects out of 2D objects, mask out other objects and background, and crop the image to the size of the object, and use CLIP encoder to encode the object. We follow ConceptGraph [3] to merge the objects from 2D to 3D. For scene construction, each object has one CLIP feature. For object details (point cloud), we project the 2D pixels of the objects to 3D, and get the point clouds of the objects.

B.2. Ablative Studies

In Table 1, we show additional experimental results where we explore MultiPLY with single, double or triple modalities.

B.3. More Qualitative Examples

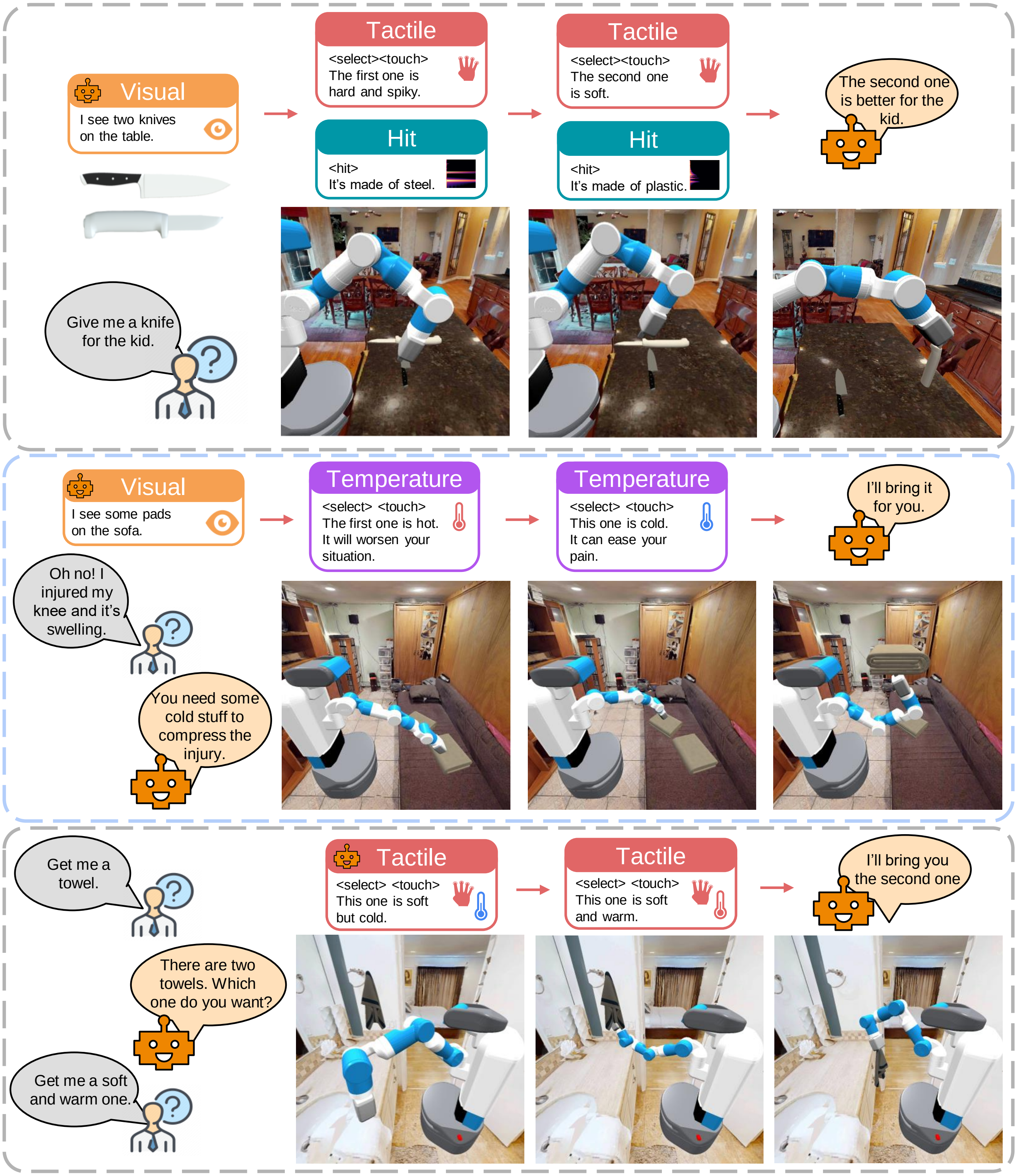

Visual
I see two knives on the table.
Give me a knife for the kid.
Tactile
<select><touch> The first one is hard and spiky.
Hit
<hit> It's made of steel.
Tactile
<select><touch> The second one is soft.
Hit
<hit> It's made of plastic.
The second one is better for the kid.
Visual
I see some pads on the sofa.
Oh no! I injured my knee and it's swelling.
You need some cold stuff to compress the injury.
Temperature
<select> <touch> The first one is hot. It will worsen your situation.
Temperature
<select> <touch> This one is cold. It can ease your pain.
I'll bring it for you.
Get me a towel.
There are two towels. Which one do you want?
Get me a soft and warm one.
Tactile
<select> <touch> This one is soft but cold.
Tactile
<select> <touch> This one is soft and warm.
I'll bring you the second one

Figure 7. More qualitative examples of MultiPLY

References

[1] Anonymous. DIFFTACTILE: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. under review. 2

[2] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. *ArXiv*, abs/2109.07991, 2021. 2

[3] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning, 2023. 5

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023. 5