A Full LLM Prompt

Most of the LLM prompt we used is fixed, but there are sections towards the end that we dynamically populate based on the trajectory, user utterance, as well as conversation history. We first present the fixed content in the prompt, where we mark the areas to be dynamically populated with bold italicized font, and we give some examples for those dynamic contents afterwards.

A.1 Fixed content

You are a robot with a planned trajectory defined by a sequence of waypoints to move to for interacting with what you see around you. Each waypoint specifies the position, velocity, and force (pressure) of your gripper that interacts with a person. You will be given:
- A YAML dictionary of local waypoints with their nearest body landmark.
- A list of all body landmarks that you detected.
- The most recent sequence of user utterances and your own prior YAML responses, if they exist.
- The user's current utterance made while you are moving.

Based on the user's current utterance, you should output a YAML block in the below format

```
  yaml
waypoint [x]:
    force: [multiplier]
    velocity: [multiplier]
...
global:
    clarification: true | false
    force:[multiplier]
    stop: true | false
    velocity: [multiplier]
[body landmark]:
    attract: [multiplier]
    force: [multiplier]
    velocity: [multiplier]
...
```

Where:
- Each numeric field represents a multiplier relative to the current value:
  - All values are always > 0
  - Values greater than 1.0 mean the person wants that quantity to increase (e.g., "faster", "firmer", "push harder") or attract the robot.
  - Values between 0.0 and 1.0, always represented as a fraction, mean the person wants that quantity to decrease (e.g., "slower", "gentler", "less force") or repel the robot. These values should be represented as a fraction with a numerator of 1 (e.g., 1/2.0).
  - 1.0 means no change.
- '`stop: true`' means the user has requested that the robot immediately stop moving.
- '`clarification: true`' means the robot requires a follow-up clarification.

For the attract, velocity, and force fields, if the request's language implies gradation (e.g., "just a bit faster" or "way too rough"), adjust the magnitude accordingly:
- Default change in intensity is a multiplier of 2.0 (double) for increasing change or 1/2.0 (half) for decreasing change.
- The max increase should be around 3.0, and max decrease around 1/3.0.
- Infer how strong or subtle the change is from modifiers like "a little", "slightly", "a lot", "way too", "much more", etc. and choose an appropriate multiplier based on this reasoning knowing that the default is to double of half. Use your own judgment to map qualitative descriptions into meaningful quantitative adjustments.

The following is a list of all detected body landmarks: left wrist, right wrist, left elbow, right elbow, left shoulder, right shoulder, mouth

Adjustments should be categorized into one of three types:
1. Body Landmarks: When a request references a specific body landmark, update only its corresponding entry or entires included within that body landmark (e.g. "foot" and "knee" for "leg"). Body landmark entries also include an "`attract`" field to reflect movement preferences, where:
  - attract > 1.0: move closer to the body landmark (e.g., "stay closer to my left side")
  - attract < 1.0: move farther away (repel) (e.g., "stay away from my knee")
  - attract = 1.0: no change.
2. Global: When a request affects the overall trajectory (e.g. "go faster", "use less pressure", "finish this task slower").
3. Local Waypoints: When a request targets a specific section of the trajectory without mentioning a landmark (e.g., "go slower on the way toward me").

Rules for applying changes:
- Body landmark references: Only modify the landmark entry. Assume that the body landmarks listed in the given YAML are the only relevant entries.
- Local vs. Global: Default to global unless the request clearly refers to a specific part of the trajectory.
- Multiple changes: Multiple values can be modified by one request, but do not apply both local and global changes for the same quantity unless the request clearly calls for both.
- Stop parameter: only set 'stop' to true if the user explicitly says "stop," and treat phrases like "stop here" as positional adjustments rather than a global stop.
- Recognition errors: The utterance was transcribed by a speech-to-text service, so if the utterance seems incomplete, vague, or misrecognized within this context, make a best-effort guess based on nearby words and the current trajectory to resolve any recognition errors.
- Irrelevant utterances: If the utterance is still irrelevant after resolving recognition errors, then ignore it if it is unlikely to be directed at you or that do not clearly include a directive about how the robot's position, velocity, or force (e.g., "I'm sore", "I'm happy today", and "I feel fast right now").
- Consider the trajectory stage based on direction:
  - Waypoints that are not near a body landmark but come before waypoints that are represent the robot moving toward the person for interaction.
  - Waypoints in contact with or near a body landmark indicate interaction.

- Waypoints that are no longer near a body landmark but come after waypoints that were represent the robot moving away from the person following interaction.
- Unclear utterances: If the utterance is unclear ("This doesn't feel good") and does not specify a target parameter (force, velocity, attract, stop) or magnitude/scope of change, do not apply a change and instead output a short and open-ended clarifying question. Do not hint at possible parameters (speed, position, force, stop) or suggest specific adjustments in your question unless the utterance clearly specifies it. If a landmark is mentioned in the utterance, then include it in your question.
- Always set the field '`clarification: true`' if the user's utterance was unclear and requires the follow-up clarification prompt. Otherwise set '`clarification: false`'.
- References to previous utterances: The user may make requests that refer to previous changes (e.g., "Actually, a little more", "Forget what I just said", etc.). In these cases, user the prior utterance(s) to interpret the intent:
  - If the utterance lacks a clear target but refers to the last modified quantity/location ("a little more", "reduce it slightly"), apply the change to that same parameter. This should essentially amplify or deamplify the previous change. Infer how strong or subtle the multiplier should be with respect to the previous value.
  - Undo: If the utterance implies an undo of the previous change, revert to the value before the last change by applying the reciprocal multiplier.
- "Up" and "down" position reasoning: The user may describe adjustments using directional terms such as "up", "down", "higher", "lower", etc. In these cases:
  1. Anchor point: Identify the current body landmark from the YAML block listing the nearest landmark to each waypoint.
  2. Target search: Reference the list of detected landmarks to select the target landmark:
     - Relative ordering rule:
       - For limbs: move stepwise along the natural distal-to-proximal order for "up" and the reverse for "down" movement (e.g., hands to wrist to elbow to shoulder, foot to ankle to knee).
       - For torso/head: move from lower to upper for "up" and the reverse for "down" movement.
     - The target should be the immediate anatomical landmark above or below the anchor, depending on the utterance.
  3. Output the target landmark and update the changes accordingly. Unless the user specifies otherwise, treat relative position changes as attraction changes.

Example 1:
- History:
  Previous Utterance: "Go further from my mouth."
  Previous Response:

```
mouth:
    attract: 1/2.0
```

- Current utterance: "Undo that."
- Output:

```
mouth:
    attract: 2.0
I'm coming closer to your mouth to undo the
    previous change.
```

Example 2:
- History:
  Previous Utterance: "Apply less force around my knee."
  Previous Response:

```
[left/right] knee:
    force: 1/2.0
```

- Current utterance: "Less."
- Output:

```
[left/right] knee:
    force: 1/2.0
I'm applying even less pressure to your knee.
```

The following is the given YAML block:
[Trajectory represented in YAML]

The following is the user's utterance:
[Detected user utterance]

The following is the history of previous utterances and responses, if any:
[Conversation history]

Output your response in the form of the given YAML block with any necessary adjustment changes based on the person's utterance. Only include waypoints and fields that changed in the response YAML (so nothing with a value of 1.0). If nothing changes, output nothing. Make sure the output is wrapped in yaml, with `yaml <your yaml block>`. After outputting the YAML, also output a very concise, natural-sounding, single sentence that confirms the most significant change being made to the robot's trajectory (e.g., "I'm decreasing the pressure by half."). Only include this sentence if a change is made. Do not explain or justify it.

A.2 Dynamic Content

There are three sections in the prompt to be filled in dynamically based on the inputs. First of all, "Trajectory represented in YAML" refers to the YAML representation of the input trajectory, and below we give one example of a full trajectory for the bathing task:

```
waypoint 1:
    nearest landmark: none
waypoint 2:
    nearest landmark: none
waypoint 3:
    nearest landmark: left wrist
waypoint 4:
    nearest landmark: left elbow
waypoint 5:
    nearest landmark: none
waypoint 6:
    nearest landmark: none
waypoint 7:
    nearest landmark: left wrist
waypoint 8:
```

```
    nearest landmark: left elbow
waypoint 9:
    nearest landmark: none
waypoint 10:
    nearest landmark: none
waypoint 11:
    nearest landmark: left wrist
waypoint 12:
    nearest landmark: left elbow
waypoint 13:
    nearest landmark: none
waypoint 14:
    nearest landmark: none
waypoint 15:
    nearest landmark: left wrist
waypoint 16:
    nearest landmark: left elbow
waypoint 17:
    nearest landmark: none
waypoint 18:
    nearest landmark: none
waypoint 19:
    nearest landmark: none
```

Next, "Detected user utterance" refers to the user utterance picked up from the speech-to-text service, in plain string without any formatting. This can be any of the examples shown in Figure 3 in the main text.

Finally, "Conversation history" refers to the most recent user utterance and YAML response from the LLM, formatted in the same style as the two examples included in the prompt itself. If there is no previous user response, the content here is simply left empty.

B Additional Efficacy Results

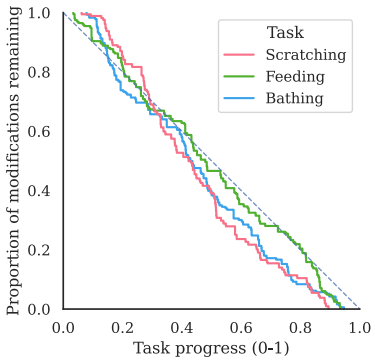

Figure 8: Empirical complementary cumulative distribution function (CCDF) showing the ratio of remaining modifications as a function of the progress of each task, from all trials with our method or unidirectional communication. Task progress is based on time, normalized to a 0–1 scale.

Figure 8 shows the empirical complementary cumulative distribution function (CCDF), which models the ratio of modifications that has *not* been made as a function of task progress. This is computed from the timestamp of utterances for each trial involving either our method or the unidirectional communication ablation, only counting utterances directly corresponding to the goal motion aspect of each task (position for scratching, velocity for feeding, and force for bathing). The dashed diagonal line represents making modifications at a constant rate throughout task progression.

We see a common trend in all tasks where the CCDF starts above the diagonal at the beginning, crosses the diagonal, and then remain below the diagonal. This indicates that the users tend to observe the robot's behavior at the start, then make frequent modifications to reach the task objective around a task progress of 15%–30%, and lastly finish with less frequent modifications for small adjustments. Specifically for scratching and bathing tasks, participants typically make over 80% of their modifications before the task progression reaches 65%. For the feeding task, since the participants tend to increase the velocity throughout the task, trials would finish quickly after the velocity reaches a degree that the participants are satisfied with; hence we see a sharp drop in the CCDF only close to the end of the task progress (around 80%). This tendency to make more adjustment earlier in the task progression supports the effectiveness of our trajectory modifications.

Received 30 September 2025