We run our tests with each possible combination of these

and a subjective command like "get ready for a party", the

three contexts and commands (9 total), each for 10 trials. We

LLM simply makes up a response-specifically, it turns on

save the agent's response for each trial in a human-readable

all the lights in the house, to include the bedroom. When

format, then perform manual rating to measure the quality of

we add a speaker and a television to the context, the model

the responses. Our process for rating the quality of responses

now has more relevant knobs to turn, and produces a higher

is based on a binary label, where each is assigned one of the

quality response. The tendency to make something up when

following labels based on its quality:

the answer is unknown or requires more context is an open

Poor (0): "The changes to the devices do not at all reflect

problem and motivates the development of application-specific

methods to mediate between the user and the model.

the intent behind the user command, or the response is

malformed/garbled."

For the most ambiguous command ("I am tired"), we note

Good (1): "The changes to the devices are reasonable for

that the model delivers poor responses regardless of context.

the command. You can imagine someone being satisfied

In all but a few cases, the LLM simply turns on all of the

with the result, even if it is somewhat subjective (e.g.,

home's lights. The exception is in a Medium/Ambiguous trial, based on different personal preferences)."

where it only turns on the bedroom lamp, perhaps to help the

user prepare for bed. This is to be expected: an individual's

Three researchers independently reviewed all responses and

intent and preference in this case are highly subjective (are

assigned them a label. We report the aggregate score for each

they, e.g., tired and ready for bed or tired but they have a

trial as the average across all assigned scores. We also note the

pressing deadline?) and the LLM ultimately cannot read the

average latency for each trial-this includes both the network

user's mind. However, since the LLM does not "know what it

transmission time of the request, as well as the inference time

doesn't know", it does not ask for clarification or go with the

taken by the model. Since this time is subject to network

safest choice, which is likely to do nothing. Instead, it makes

conditions and API demand, it should be taken as a rough

something up.

estimate rather than a concrete benchmark. Our results are

Since our previous results suggested more context is often

summarized in Table

beneficial, we dig deeper to see if we can help it make a better

Response time is a function of context complexity. With

choice. We amend the vague command "I am tired" to offer

respect to latency, we can see that responses generally arrive

hints at the user's intent:

on the order of seconds, meaning that a practical system

Ambiguous* "I am tired and I need to work."

could feasibly leverage an LLM for ambiguous command

Ambiguous "I am tired and I want to sleep."

inference and action planning without significant detriments

to user experience or responsiveness. For direct commands, a

Provided this added hint at the user's context, the response

2 to 3 second response time may be too long-future system

quality improves significantly. For Ambiguous*, the model

designs could thus leverage the LLM only for commands that

consistently responds by turning on the living room lights

require it. This may entail a hybrid of rule-based inference

while leaving all other devices off; for Ambiguous**, the

for common commands, along with LLM inference for less

model turns on only the user's bedside lamp and, in some

familiar commands. It is also worth noting that as the context

cases, reduces the volume on their speaker and TV. Note that

increases in complexity, the response latency also increases-

although the amended statement includes additional context,

this motivates future work to develop methods for filtering

it still requires the model to infer meaning in a way that more

context prior to prompting the agent SO that only the most

rigid or rule-based approaches cannot.

relevant information is provided.

V. IMPLEMENTATION

Response quality is a function of context and command

ambiguity. With respect to response quality, we find that the

To demonstrate LLM-driven smart home control in prac-
LLM approach provides good responses given the same direct
tice, we built a proof-of-concept implementation in Python.
and simple commands that current home assistants are able to
Our implementation accepts user commands as strings, pack-
service. Note, however, that unlike existing home assistants,
ages them into prompts along with contextual information
the LLM approach utilizes a much simpler system architecture
about real devices, then processes responses from OpenAI's
that performs command inference and action planning in
text-davinci-003 model into smart home API calls that
the same pass. These results are consistent given increasing
change the device state as specified by the LLM. We scope
degrees of context complexity, suggesting that the model was
the application to one room (a researcher's living room) with
not overwhelmed by the growth in the decision space that
three Philips Hue color smart lights 19 and one TP-Link
comes with adding new devices and possible state changes. On
Kasa smart plug [26] that controls a stereo. We store the
the contrary, the model provides better responses when given
device context in JSON as in our experimental setup, with
more context that might be relevant to the user's command,
the difference that the device state for the Hue lights is pulled
as is apparent when comparing the low response quality of
directly from the Hue API without modification. Our code is
the Simple/Indirect and Medium/Indirect experiments against

open source and available online

the Complex/Indirect experiment. Upon inspection of the

responses, the reason for this is clear: given minimal context

https://github.com/UT-MPC/homegpt