lapsed and the dialog history, this method offers more scene level visual context to the LLM.

Object and State Detection (Obj): In this method, we extract more fine-grained scene information by detecting important objects in the frame image and their corresponding states, and inserting them into the prompt observations. At each query time t, we use EgoHOS (Zhang et al., 2022) to segment user's hands and the objects that they are interacting with. We then extract the object segments and predict their most likely object names and their corresponding states using CLIP (Radford et al., 2021b). To balance the performance and generalizability of CLIP predictions, we first extracted a list of most likely objects and their potential states from the recipe through a separate prompt to the LLM backbone. The resulting list of objects and states go through a minor manual cleanup before being used by CLIP. This narrows down the scope of search for CLIP, and offers better targeted vision to language prediction, but is also easily generalizable to open-world object and state detections.

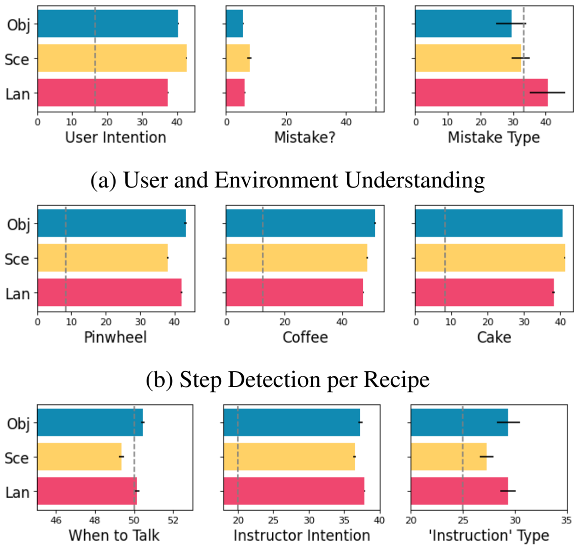
All three methods were queried at the same time points as extracted in Section 4 for fair comparison. We reserved 6 recordings (2 of each recipe) for hyperparameter and prompt tuning, and the rest for evaluation. Each method was repeated 3 times. ChatGPT APIs configurations are: Max tokens=100, temperature=0, stop words= [\n \n, —, """, ”’]. All experiments conducted on a single GPU NVIDIA RTX A6000 and a Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz.

6 Experimental Results

In this section, we evaluate the three methods above on the following two tasks: *User and Environment Understanding*, and *Instructor Decision Making*. Micro F1 scores are reported for each classification task. A total of 5921 data points were evaluated based on the three query conditions (Section 4). We further break down the performance, and evaluate how accurately each vision extraction module can translate the scene into language, and how helpful or annoying the model's generated sentences are under a human evaluation.

6.1 User and Environment Understanding

As an interactive task guidance agent, it is important for the model to have a comprehensive understanding of the task's physical environment, as well as users' mental and physical states, through their conversations as well as actions.



(a) User and Environment Understanding

(b) Step Detection per Recipe

(c) Instructor Decision Making

Figure 5: Interactive Task Guidance Micro F1 Scores: For user and environment understand tasks, the models demonstrated well above random chance performance in user intention prediction and step detection, but struggle with mistake recognition. For Instructor decision prediction tasks, the models showed above random chance performance in predicting instructor's intention and instruction types, but issued higher communication frequencies than human instructors. Across the three methods, Language Only (Lan) showed comparable performance even without any visual context.

The overall user utterance intention prediction, step detection, mistake recognition and type prediction performances can be found in Figure 5a,5b. It was observed that all three methods using zero-shot foundation models have demonstrated decent performance significantly above the random guessing (grey dash line) on user intention predictions and step recognition, but struggled with mistake detection. This is most likely due to the limited visual context the models can offer to accurately detect mistakes (More in Section 6.3). Out of the ones that the model did correctly predict that a mistake has happened, it displayed chance level of performance.

Among the three methods, it is observed that with just the dialog context alone, the model was able to achieve comparable performance as the other two. This shows that the conversations between the human user and instructor disclose a lot of information about the user and the environment even without visual perception inputs. The Object