Connect the Dots: Zero-Shot Step Detection with Foundation Models

Anonymous ARR 2025 submission

Abstract

When performing a task such as making a cup of coffee or replacing a bicycle tyre, an individual or 'User' might seek further guidance to ensure that the task is completed correctly. Foundation models are suitable candidates to provide this guidance automatically. However, a model must first be able to grasp a given situation to provide situated guidance. This work focuses on 'Step Detection' (SD), where a model is asked to detect which step of a task a User is performing given a dialogue history and an image of the current scene. We leverage open-access language and vision-language foundation models to perform zero-shot SD on the Watch, Talk and Guide benchmark. We show that current publicly available models achieve up to 54.40 F1, outperforming ChatGPT-3.5 by 12%. To enhance the performance of VLMs on SD, we propose to apply 'structured Chain of Thought (CoT)'. This approach guides the model through a multi-turn interaction to steer it to the correct answer. We demonstrate that structured CoT can lead to significant improvements when scene images are clear and relevant. We also demonstrate that leveraging predictions from an image classifier trained on in-domain data yields further performance gains.

1 Introduction

As large-scale foundation models rapidly develop, we begin to consider their applicability to tasks that require knowledge of the physical world. Recent multimodal foundation models such as Gemini 2.0 (Georgiev et al., 2024) and GPT-4 (Achiam et al., 2023) have been able to demonstrate increasingly complex capabilities across multiple modalities, making them suitable candidates to form the backbone of next-generation virtual and embodied assistants. However, to serve as reliable assistants these models must first be able to perceive a real-world environment and interpret the current situation. One specific aspect of situational understanding is detecting which step of a procedural task an individual is performing. This step detection problem is one that requires models to relate various sources of information, such as the immediate scene or a short video leading up to the scene, an individual's previous utterances and prior (potentially pre-trained) knowledge of the task at hand.

Much of the work in the area of virtual assistants is conducted in well-controlled virtual environments where interactions are often simulated. While this is practical and cost effective, it remains useful to benchmark models against real-world settings to obtain a more realistic account of how they would perform in practice. The recently released Watch, Talk and Guide (WTaG) benchmark (Bao et al., 2023) provides first-person videos of individuals performing relatively simple kitchen tasks alongside transcriptions of their real-time interactions with a human instructor. This dataset provides a suitable sandbox to investigate different strategies of performing zero-shot step detection with foundation models. So far, only ChatGPT 3.5 has been evaluated on WTaG (Bao et al., 2023). We believe that benchmarking more recent publicly available multimodal foundation models would be useful to the wider community, not only to assess the readiness of these models in real-world applications but also to improve the accessibility and reproducibility of our experiments.

This work makes two key contributions, both with the aim of extracting useful insights for building practical applications with foundation models. The first is a focused evaluation of a range of open access foundation models on the Step Detection (SD) sub-task of the WTaG benchmark. We choose this task because it requires processing of both video and transcript, making it an ideal task-oriented benchmark for multimodal foundation models. We study two methods of combining information from different modalities, namely captioning and direct multimodal processing. Our experiments show that captioning generally leads

to better results than direct multimodal processing. Moreover, our results support the intuition that selecting a relevant, representative image of the current scene is central to achieving strong performance.

Our second contribution is to enhance SD performance on WTaG by applying 'structured' CoT as an alternative method to standard CoT. Instead of prompting a model in the usual way, structured CoT conducts a guided interaction consisting of static prompts in an attempt to steer the model to systematically deduce the correct answer. We find that structured CoT can indeed yield gains in performance, *provided that there is already a relevant image of the scene available*. We also demonstrate that these gains are sensitive to task complexity. Finally, we extend the CoT setting to incorporate probability distribution outputs from an in-domain image classifier, and observe further improvements. This suggests that strong assistive agents can potentially be bootstrapped by composing a simple system like an image classifier with a powerful VLM.

In the following sections, we introduce the WTaG benchmark (Section 2) and the experimental setup we adopt to produce results for open-access models on this benchmark (Section 3). We consider using standard LLMs, the same LLMs equipped with a caption of the current scene and VLMs that can directly process multimodal inputs. In Section 4, we elaborate on the proposed structured CoT approach to improve SD performance. The results and the corresponding discussions from all our experiments are subsequently covered in Section 5.

2 Dataset

The WTaG dataset consists of 48 publicly available first-person videos in a real (i.e. non-simulated) environment. The videos show a User following a recipe to make either a pinwheel pastry (similar to a peanut butter and jelly sandwich), a cup of coffee or a small cake. The videos range from 5 to 18 minutes in length. The median video length is 10 minutes. In total, the dataset contains 4,233 English dialogue utterances (Bao et al., 2023).

Throughout a video, the User is verbally guided by a human Instructor. Both User and Instructor utterances are manually transcribed and the transcription is aligned with the video. The WTaG benchmark is constructed by extracting a set of 'query points' (QPs) from these videos. A QP can be thought of as a situational snapshot at a point in time that is used to make a prediction (step of the recipe, whether to speak, etc.). The features associated with this point, including the text transcript of the interaction, and visual information, can be used as input to the model that makes a prediction. For each QP, the LLM/VLM is queried with a prompt that broadly follows the same structure as the template described in Bao et al. (2023). This template includes the recipe of the task the User is performing as well as the running dialogue history in order to contextualise the current situation. Following Bao et al. (2023), three different types of QPs are extracted whenever one of the following conditions is satisfied: 1) the User speaks, 2) the Instructor speaks and 3) Neither User or Instructor have spoken for 10 seconds.

	Pinwheel	**Coffee**	**Cake**	**Total**
User	501	567	930	1998
Instructor	429	472	698	1599
Wait	490	599	581	1670
Total	1420	1638	2209	**5267**

Table 1: Distribution of QPs across the WTaG benchmark across task and query type.

Extracting QPs across all the raw videos results in the final WTaG benchmark that consists of 5267 total QPs, roughly distributed across the 3 tasks. Table 1 shows the exact breakdown of the QPs in the benchmark. In the full benchmark, a foundation model can in principle be evaluated against 7 tasks such as 'User Intent Prediction', 'Mistake Detection', or 'Step Detection'. In this work, as a first step towards a situational assistant, we focus on Step Detection, which relies both on information from the video and dialogue history. Instructor and User Intent Prediction for instance rely less on visual information and can be optimised with text information alone.

Formally, we define a query for a QP Q_i as either a text prompt $\mathbf{H_L}$, which includes the dialogue history up until the QP occurs, or a tuple that contains $\mathbf{H_L}$ and a paired image $\mathbf{H_V}$, which is a still frame from a WTaG video. We do not consider multiple frames due to compute requirements and our initial observations that current publicly accessible video LLMs such as Video-LLaMA (Zhang et al., 2023a) and LLaVA-NeXT Interleave (Li et al., 2024a) do not yet have strong reasoning skills compared to standard VLMs.

$$Q_i = \begin{cases} (\mathbf{H_L}, \mathbf{H_V}) & \text{if } \mathbf{H_V} \neq \text{null}, \\ \mathbf{H_L} & \text{otherwise.} \end{cases}$$

Suppose we are given a QP Q_i, an autoregressive foundation model P_θ and a description of a physical task with K steps. The task of P_θ is to classify Q_i to the correct step k. This is referred to as 'Step Detection' in Bao et al. (2023). We then evaluate P_θ across the 3 tasks in WTaG. For reference, the pinwheel and cake tasks consist of 12 steps, while the coffee task consists of 8.

3 Evaluating Open-Access Models on Step Detection

We run a range of experiments to benchmark several foundation models on the SD task in WTaG. We extend the original evaluation framework [1] to handle more models and experiment setups. These experiments are designed to evaluate current open-access models against the closed-source ChatGPT-3.5 baseline and to yield practical insights about how foundation models combine multimodal information. In particular, we implement the following experimental setups:

- **Blind LLM**, where a model P_θ only has access to the textual context $\mathbf{H_L}$ of a QP.
- **Blind LLM w/ Caption**, where P_θ is a language model augmented with a text caption representing $\mathbf{H_V}$.
- **Single-Frame VLM**, where P_θ is a multimodal model that can natively handle text-image pairs $(\mathbf{H_L}, \mathbf{H_V})$.

The above nomenclature is loosely adopted from Majumdar et al. (2024) and (Zeng et al., 2022). We consider two families of open access models, based on the popular LLaMA-2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023) models. For the LLaMA-based models, we consider the 7B and 13B instruction-tuned variants of LLaMA-2, as well as Vicuna (Chiang et al., 2023) and LLaVA-NeXT (Liu et al., 2024a). Vicuna is initialised as the base LLaMA-2 model but is further fine-tuned on instruction-following data generated by ChatGPT, procured through the ShareGPT platform[2]. LLaVA-NeXT is a VLM that is a successor to the popular LLaVA model (Liu et al., 2024b) that uses Vicuna as its language backbone. We also consider the recently released Pixtral model (Agrawal et al., 2024), which is based on a bespoke vision encoder Pixtral-ViT and the Mistral-NeMo LLM, designed to natively handle images of different resolutions and aspect ratios. The parameters of all the models are frozen across all experiments.

For the **Blind LLM w/ Caption** setup, we consider Pixtral and Vicuna as the blind models to be augmented. When captioning $\mathbf{H_V}$, we adopt a 'self-captioning' approach. For Pixtral, this means that we run the model with text-only input, but augment this input with a caption generated by running the model separately as a Single-Frame VLM. Because Vicuna is a text-only model, we instead append captions to the default prompt template. The captions are generated by LLaVA-NeXT, as Vicuna forms the backbone for this model.

Importantly, for the settings where an image is required i.e. **Blind LLM w/ Caption** and **Single-Frame VLM**, we examine two scenarios. The first is the default scenario described in Bao et al. (2023), where $\mathbf{H_V}$ is selected simply as the video frame corresponding to the timestamp at which a given Q_i occurs. Under this approach, image quality can be volatile; frames could be blurry, irrelevant or unhelpful to the model since the User could turn their heads, get distracted and focus their gaze on a region of the scene that is unhelpful to the Instructor, etc. Thus, we examine a second scenario, 'fixed image', where we hand-pick K frames (one representative of each step in a task) that are clearly discernable by humans and use these as the accompanying image to $\mathbf{H_L}$. Under this setting, many QPs will share the same visual information, but differ in their text components. We consider this setting an upper bound on performance as we effectively guarantee that the models see a 'clear and representative' image. In real-world applications, efficient algorithms to select meaningful frames would be necessary. Note that we could not run exhaustive combinations of experimental settings due to constraints on compute.

4 Structured Chain of Thought

To further improve SD performance for VLMs we explore structured Chain of Thought (CoT) (Wei et al., 2022). Under standard CoT, rationales are generated in one step, usually with the command 'think step by step', making each intermediate rea-

[1] https://github.com/sled-group/Watch-Talk-and-Guide
[2] https://sharegpt.com/

soning stage implicit and uncontrollable. Models are also expected to have emergent few-shot in-context learning abilities which have been shown to be more limited in smaller scale models (Brown et al., 2020).

In the structured CoT approach we propose, rationales are instead generated iteratively, through a multi-turn interaction where at each turn, the model is prompted to reason by looking at the output of the previous turn and the provided evidence (see Figure 1). The exact prompts are included in Appendix A.2. These prompts attempt to simulate how a human would reason about a multimodal situation and 'connect the dots' in order to systematically deduce the correct answer. The hypothesis is that this explicit reasoning chain will allow us to steer the model towards the correct answer. Through our preliminary experiments, we observed that LLaVA-NeXT does not have the base reasoning capabilities to benefit from this approach. Thus, we opt to study Pixtral only for these experiments.

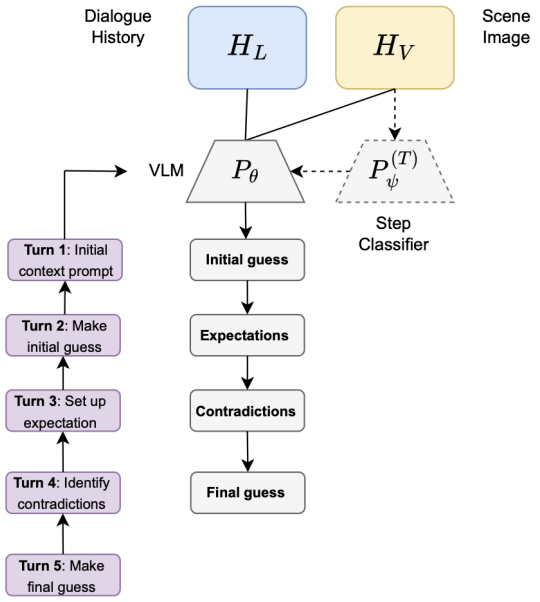


Figure 1: Schematic diagram of structured CoT. The step classifier $P_\psi^{(T)}$ is an optional component of the setup and is shown with dotted lines.

The deductive process begins with a prompt to contextualise the task (this is similar to the prompt used in the original WTaG setup) and another to record the model's initial guess, $\hat{k}_0$. Given $\hat{k}_0$, D follow-up prompts are passed. These are designed to elicit a set of expectations from the model based on its initial guess. Then, the model is instructed to identify any evidence in the given scene that *contradicts* its initial expectations. The idea is to elicit the model to update its current guess if the evidence it has been provided is inconsistent with its expectations.

4.1 Leveraging In-Domain Classifiers

While structured CoT has potential to improve performance, we also investigate the direction of leveraging bespoke image classifiers for the SD task. In this case, we train a set of in-domain classifiers for each (physical) task in WTaG, and use their outputs for zero-shot step detection with a VLM. The idea here is to steer the model towards steps that are deemed more probable by an external expert model. In other words, we can think of this process as biasing the prior knowledge of the model with task-specific knowledge of which steps are more or less likely given the current scene. Under this setup, the in-domain classifiers encode domain-specific knowledge about the WTaG tasks, which the VLM should be able to exploit to improve performance.

Formally, our objective is to train a simple image classifier $P_\psi^{(T)}(k|\mathbf{H_V})$, where T is one of the WTaG tasks. We choose a standard ViT (Dosovitskiy, 2020) as $P_\psi^{(T)}$. However, once we have obtained this distribution the question becomes: how should $P_\psi^{(T)}$ be leveraged by the foundation model, P_θ? In this work, we opt for the cheap-and-cheerful method of encoding the classifier distribution in the setup prompt as a string (see Appendix A.3). We do not pass the raw probabilities but rather ordinal categories (i.e. Very Low, Low, Medium, High, Very High) as we found that Pixtral struggled with comparing numerical values. These categories are split at uniform intervals.

5 Results

This section summarises the key findings from our experiments. Regarding implementation details, we extended the WTaG evaluation framework to conduct our experiments. Experiments were run on a single A100 GPU, or 4 V100 GPUs. To train the in-domain classifiers, we subsampled 1 fps from the WTaG videos and obtain datasets of 7509, 8011 and 9546 images for the pinwheel, coffee and cake tasks respectively. We then fine-tuned 3 vanilla ViT models with LoRA (Hu et al., 2021) via the AdapterHub framework and use the recommended hyperparameters for image classification (Poth et al., 2023) (see Appendix C for details).

Model Type	Model	Model Size	Pinwheel F1	Coffee F1	Cake F1	Overall F1
	ChatGPT-3.5	-	42.09	47.27	38.23	42.53
Blind LLM	Vicuna	13B	31.06	53.54	50.38	44.99
	LLaMA-2	7B	29.37	19.11	23.90	24.13
	LLaMA-2	13B	26.27	45.91	36.76	36.31
	Mixtral	47B	38.45	47.07	55.32	46.95
	Pixtral	12B	42.32	53.17	60.25	51.91
	ChatGPT-3.5+BLIP-2	-	37.99	48.64	41.75	42.79
Blind LLM w/ Caption	Vicuna+LLaVA-NeXT	13B	28.59	50.18	49.75	42.84
	Pixtral+Pixtral	12B	45.00	56.47	61.74	54.40
	Pixtral+Pixtral (Fixed image)	12B	**58.59**	**56.78**	**65.91**	**60.43**
	LLaVA-NeXT	13B	28.73	53.79	51.34	44.62
Single-Frame VLM	LLaVA-NeXT (Fixed image)	13B	30.63	55.31	53.06	46.33
	Pixtral	12B	38.80	55.07	55.82	49.90
	Pixtral (Fixed image)	12B	55.21	54.40	59.17	56.26

Table 2: Summary of key results from the various benchmarking experiments we conducted. Best F1 scores among the open access models (regardless of type) are shown in bold. The ChatGPT-3.5 results are as reported in Bao et al. (2023).

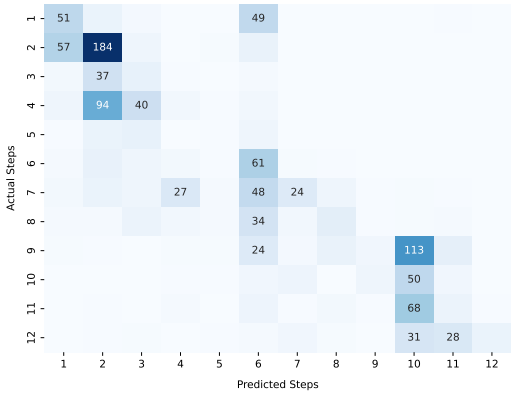
Training was seeded and took about 20 minutes for each classifier. Inference was much more expensive, with some experiments taking up to 20 hours. As a result, the results presented for each task consist of single runs. The mean of these scores are then presented as the overall F1 for that model and experiment setup.

5.1 Language-Only Baselines

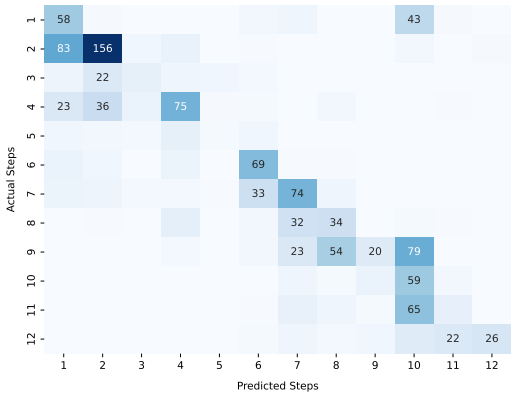
From Table 2 we observe that the **Blind LLM baselines are strong**, with Vicuna, Mixtral and Pixtral outperforming the reported ChatGPT-3.5 results. This demonstrates that dialogue cues alone provide significant information for models to detect the step. This observation agrees with similar findings from Majumdar et al. (2024).

Further, we find that certain steps in a task are more strongly signalled than others. Figure 2 shows that Steps 2, 6 and 10 of the pinwheel task are well signalled and are often correctly classified by models. We call such steps 'landmark' steps. The risk of landmark steps is that they are 'sticky', meaning a model may fixate on them, misclassifying the following steps (as seen in Figure 2a). Pixtral's superior performance can thus be attributed to its ability to identify more 'in-between' steps and mitigate the risk of fixation on landmarks as seen in 2b where steps 4, 7, 8 and 9 are better classified by the model.

Regarding scale, LLaMA-2 7B is outperformed by all larger models, suggesting that SD is difficult and requires a certain base level of reasoning to perform well. However, Pixtral exhibits higher F1 scores than both ChatGPT-3.5 and Mixtral (assuming the former has a similar size to GPT 3 (Brown et al., 2020)). This shows that models also do not need to be exceedingly large to perform SD; a medium-sized model with a strong language backbone (in this case Mistral-NeMo) is sufficient.



(a)



(b)

Figure 2: Confusion matrices of Vicuna 13B (top) and Pixtral (bottom) for the pinwheel task, using only the dialogue history as context.

5.2 Effectiveness of Scene Captions

As for the **Blind LLM w/ Caption** setup, we obtain the intuitive results that **1) image information can improve SD performance and 2) caption quality is important for achieving strong SD performance**. Captions generated with LLaVA-NeXT consistently degrade the relative base 'blind' performance of Vicuna 13B (see Blind Vicuna v. Vicuna+LLaVA-NeXT), while the captions generated with Pixtral lead to gains relative to the blind performance for all 3 tasks (see Blind Pixtral v. Pixtral+Pixtral).

Table 2 also demonstrates that captioning generally performs better than processing the scene information jointly with the dialogue history, as the scores for Single-Frame VLM are lower than the scores of LLM with caption. In other words, using P_θ to map $\mathbf{H_V}$ to language space first is more helpful than passing $\mathbf{H_V}$ as is to the model and predicting the target step in the following way:

$$\hat{k} = \arg\max_{k \in K} [P_\theta(k|\mathbf{H_L}, \mathbf{H_V})]$$

Where $\hat{k}$ is the predicted step. We hypothesise that the drop in performance on the **Single-Frame VLM** setting is due to 1) the models struggling to focus on aspects of the full scene that are relevant to inferring the current step, 2) the models not having enough context to deduce the current step and 3) excessive hallucinations i.e. 'seeing' the scene inaccurately, leading to derailed rationales and incorrect predictions. While captioning is also prone to hallucination, it would seem that models can more easily attend to relevant information and disregard irrelevant cues when the given scene is represented as text.

5.3 Representative Fixed Images

The results obtained from the 'fixed image' experiments demonstrate that passing a representative image of a step to a model is crucial to achieving strong performance on SD. Across all models, experimental setups and tasks, **the 'fixed image' setting generally leads to improved SD performance**, with the Pixtral+Pixtral (Fixed image) setup performing the best overall. We also find that F1 scores increase the most on the pinwheel task when fixing $\mathbf{H_V}$ (consider the 14.59% relative gain for Pixtral+Pixtral), showing that this task is highly sensitive to accurate visual information, for reasons related to task complexity discussed in Section 5.6.

5.4 Structured Chain of Thought

Table 3 shows our results for the structured CoT experiments with Pixtral. F1 scores are computed after the model's initial guess and after its final guess. This is done to determine the effects of both the visual information and the deductive process on overall performance. It should be noted that since the initial context prompt for these experiments differs slightly from the previous experiments, the 'initial guess' results do not exactly match the blind scores of Pixtral in Table 2.

Previously we have shown that using an LLM with captioned image information outperforms VLMs that process the raw image jointly with the dialogue history, even when this scene is expected to be informative as in the 'Fixed image' setting. We find that performing structured CoT addresses these weaknesses. Under the 'Naive image' setting we observe that CoT hurts overall performance, but with the ideal 'Fixed' image it increases significantly, thus demonstrating the importance of selecting the right image when performing the SD task.

The **Socratic VLM** results show that structured CoT leads to large performance gains on the pinwheel and coffee tasks, given that $\mathbf{H_V}$ is relevant to the task at hand. Interestingly, scores for the cake task improve marginally, but not as much as the captioning setup, which yielded a score of 65.91%.

5.5 In-Domain Classifiers for Structured CoT

The trends observed for the **Socratic VLM** results are even more pronounced for the **Socratic VLM w/ Classifier** setting. Between the initial blind guess and the final guess, the scores on the pinwheel and coffee tasks improve by 22.75 and 25.4 absolute points respectively. These improvements suggest that Pixtral was successfully steered by the task-specific knowledge of the classifier to improve SD during the deductive process. It should be noted that the classifier for the coffee images performed best on the SD task and thus yielded the most gain for the VLM since it provided more accurate step distributions at each QP.

5.6 The Effect of Task Complexity

Interestingly, we observe that the scores for the cake task do not increase significantly with the deductive process, or even when leveraging the outputs from the in-domain image classifier. This

Model Type	Prompt Type	Pinwheel F1	Coffee F1	Cake F1	Overall F1
Socratic VLM	Initial guess	45.99	48.78	61.57	52.11
	Final guess (Fixed)	60.28	60.62	**63.51**	61.47
	Final guess (Naive)	46.76	48.17	58.17	51.03
Socratic VLM w/ Classifier	Initial guess	46.48	49.45	61.11	52.35
	Final guess (Fixed)	**69.23**	**74.85**	61.79	**68.62**

Table 3: F1 scores for the structured CoT or 'Socratic' experiments for Pixtral 12B, assuming the image of the scene $\mathbf{H_V}$ is hand-selected. The best VLM scores are in boldface. The 'naive image' results are included for reference.

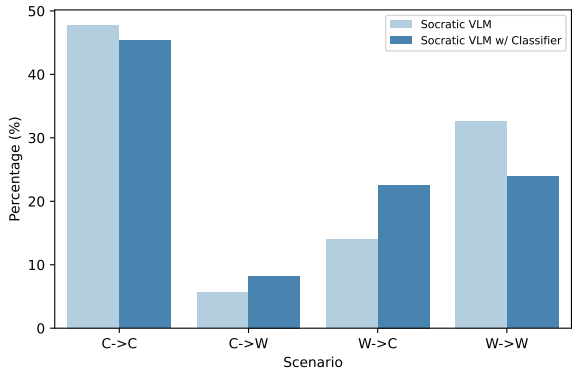


Figure 3: Breakdown of the proportions of reasoning chains (out of the total 5267) that were fixed or derailed by the different structured CoT experimental setups. 'C' stands for Correct and 'W' stands for Wrong. For example, 'W->C' means that the initial guess was wrong, but the final guess was correct.

indicates that the deductive process benefits certain tasks more than others.

To understand why this is the case, we consider the cake task in more detail. According to Table 1, the cake task has the most utterances overall. One explanation for this is that making a cake is a more complex task than making a pinwheel (similar to a peanut butter sandwich) or a cup of coffee. As such, people are less likely to be familiar with the intermediate steps of the cake task. This results in more User utterances (and by extension a lower proportion of Wait QPs) because they will seek more guidance from the Instructor, thus reducing the utility of visual information since the dialogue history contains sufficient information to determine the step. In fact, at times this visual information could risk causing a model to 'overthink', where the gains from multi-turn reasoning diminish and instead confuse the model, leading to lower performance. Comparing the Pixtral+Pixtral (Fixed image) and Socratic VLM F1 scores for the cake tasks, we find that while both lead to improvements over the text-only baselines, captioning slightly outperforms structured CoT, which we believe is due to overthinking.

Conversely, the pinwheel and coffee tasks have fewer dialogue utterances (as evidenced by a higher proportion of Wait QPs) since Users do not have to ask for guidance as often. Therefore, these tasks are inherently more reliant on the scene information. Since structured CoT is designed to make the most out of this information, we obtain the substantial performance gains observed in Table 3. In sum, we believe that the lack of improvement in F1 scores for the cake task despite using structured CoT is due to the abundance of dialogue cues (as a result of task simplicity) that makes scene information more redundant and less useful to the model.

5.7 Does Structured CoT Derail, Correct or Reinforce the First Guess?

Finally, we further analyse the reasoning chains generated by Pixtral when undergoing structured CoT. Using the initial and final guesses of the model, we measure the rates at which the deductive process is helpful or harmful to the final prediction. Specifically, we compute the percentage of Pixtral's reasoning chains (one for each QP) that fall under one of four scenarios: either the first and final guesses are both correct or wrong, or one of either the first and final guesses are correct while the other is wrong. From Figure 3 we observe that including the step probabilities from the task-specific classifiers in the deductive prompts has a slightly higher risk of derailing a reasoning chain than when these probabilities are omitted (see the 'C->C' and 'C->W' scenarios). However, this trade-off is acceptable given that the step probabilities *correct* a much larger proportion of reasoning chains (over 20%). The performance gain for **Socratic VLM w/ Classifier** can therefore be attributed to the reasoning chains that are corrected as a result of the information provided by the image classifiers. Pixtral's ability to self-correct is

encouraging, as self-correction is key for making future agents that are robust to reasoning errors and the unpredictability of real-world environments.

6 Related Work

6.1 Step Detection as Embodied Question Answering

As previously established, step detection can be performed as a text-only or vision-language task. SD is essentially a procedural segmentation task (Zhou et al., 2018), but it can also be framed as an (E)mbodied QA (EQA) problem, where an agent is tasked with answering a question that requires egocentric perception of an environment. In our case, the model is asked which step of a task a User is actively performing. Much work has been done on EQA (Bohus et al., 2024; Das et al., 2018; Schoonbeek et al., 2024; Yu et al., 2019; Li et al., 2024b), though a recent major contribution in this area is OpenEQA (Majumdar et al., 2024). OpenEQA is a large-scale benchmark that covers a broad range of questions about realistic environments that extend beyond step detection. However, they focus on multi-frame VLMs (i.e. VLMs that can handle video) rather than single-frame VLMs where only one frame of the scene is passed. We also study the utility of structured CoT rather than scene-graph captions. Another closely related work is SuccessVQA (Du et al., 2023). This work studies whether VLMs can serve as reward models by leveraging Flamingo (Alayrac et al., 2022) as a binary classifier to determine whether or not an action was completed successfully. However, SuccessVQA does not consider the setting where the guidance system has access to dialogue history, nor does it study the impact of CoT on classification performance.

6.2 Chain of Thought for Vision-Language Reasoning

Regarding vision-language reasoning, there have been substantial efforts in this direction (Amizadeh et al., 2020; Hu et al., 2024; Xu et al., 2024a; Zeng et al., 2022). Zhang et al. (2023b) in particular tackle the popular ScienceVQA benchmark by generating CoT rationales that exploit the multimodal information in a given problem, thereby grounding a model and providing hallucinations that should be less prone to hallucination. In this work we instead study *structured* CoT, where the model is instead guided through a fixed reasoning chain rather than generating the rationale itself. We believe that manually injecting this human 'know-how' of combining multimodal clues can lead to more consistent and explainable rationales. C4MMD (Xu et al., 2024b) adopts a similar approach as well, but instead applies the technique to the problem of metaphor detection, which is a binary classification task rather than the multiclass tasks we examine. Another related work is Inner Monologue (Huang et al., 2022), which explores the use of an LLM as a planner for an embodied agent. As part of the evidence provided to the planner, Inner Monologue leverages a small success detection model that first determines the likelihood that the current action was completed before passing this information to the agent. This is similar to the 'step classifiers' that we trained for the WTaG tasks.

7 Conclusion

In conclusion, this paper studied a range of methods of combining visual and text information to perform zero-shot step detection with open access foundation models. The aim was to extract insights for building task guidance agents powered by such models. First, we found that dialogue cues are crucial to correctly detecting the current step, since the text-only baselines performed well alone. Next, we empirically observed that image captioning generally outperforms direct processing, even when a relevant image of a given scene is provided.

We found that structured CoT successfully guided Pixtral to make full use of the visual information available and led to substantial gains in F1. We also observed an interesting interaction between performance and task complexity, in that tasks that are more complex may cause the User to ask for more guidance and render visual information less useful. This effectively limits our ability to generalise structured CoT to other tasks. However, the fact that Pixtral was able to leverage the outputs of the in-domain classifiers is promising because it indicates that foundation models can form the basis for what so-called 'modular agents'. Such an agent would consist of a core foundation model acting as an orchestrator to bootstrap a set of smaller, specialist models. For future work we believe it would be interesting to build a complete modular agent for this benchmark that performs all the sub-tasks at once.

8 Limitations

While we have shown that standard VLMs can perform reasonably well on the SD task of the WTaG benchmark, we focused on single-frame VLMs in this work. This makes image selection a challenge as we demonstrated that SD performance is sensitive to image quality. Ideally, we would compare our VLM results with a video-LLM baseline. However, as mentioned previously this requires extensive compute, along with sufficiently capable models that we observed were lacking. The second limitation is that the structured CoT method we propose for SD is static. In future it would be useful if this process could be automatically optimised. The fixed reasoning chain yields performance gains but there is no guarantee that it is the 'best' series of prompts that could elicit the most accurate step predictions from Pixtral. Directions such as a dual-LLM approach, where one LLM generates reasoning prompts and the other performs SD, would be interesting methods of addressing this limitation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, et al. 2024. Pixtral 12B. *arXiv preprint arXiv:2410.07073*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: A Visual Language Model for Few-Shot Learning. *Advances in neural information processing Systems*, 35:23716–23736.

Saeed Amizadeh, Hamid Palangi, Alex Polozov, Yichen Huang, and Kazuhito Koishida. 2020. Neuro-Symbolic Visual Reasoning: Disentangling 'Visual' from 'Reasoning'. In *International Conference on Machine Learning*, pages 279–290. Pmlr.

Yuwei Bao, Keunwoo Yu, Yichi Zhang, Joyce Chai, et al. 2023. Can Foundation Models Watch, Talk and Guide You Step by Step to Make a Cake? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12325–12341.

Dan Bohus, Sean Andrist, Yuwei Bao, Eric Horvitz, and Ann Paradiso. 2024. "Is This It?": Towards Ecologically Valid Benchmarks for Situated Collaboration. In *Companion Proceedings of the 26th International Conference on Multimodal Interaction*, pages 41–45.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Alexey Dosovitskiy. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, et al. 2023. Vision-Language Models a s Success Detectors. *arXiv preprint arXiv:2303.07280*.

Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context. *arXiv preprint arXiv:2403.05530*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Hanxu Hu, Simon Yu, Pinzhen Chen, and Edoardo M Ponti. 2024. Fine-Tuning Large Language Models with Sequential Instructions. *arXiv preprint arXiv:2403.07794*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024a. LLaVA-NeXT-Interleave: Tackling Multi-Image, Video, and 3D in Large Multimodal Models. *arXiv preprint arXiv:2407.07895*.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. 2024b. Embodied Agent Interface: Benchmarking LLMs for Embodied Decision Making. *arXiv preprint arXiv:2410.07166*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Zhang, et al. 2024a. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual Instruction Tuning. *Advances in Neural Information Processing systems*, 36.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, et al. 2024. Openeqa: Embodied Question Answering in the Era of Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498.

Clifton Poth, Hannah Sterz, Indraneil Paul, et al. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore. Association for Computational Linguistics.

Tim J. Schoonbeek, Tim Houben, Hans Onvlee, Peter H.N. de With, and Fons van der Sommen. 2024. IndustReal: A Dataset for Procedure Step Recognition Handling Execution Errors in Egocentric Videos in an Industrial-Like Setting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4365–4374.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. LLaMA 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024a. LLaVA-o1: Let Vision Language Models Reason Step-by-Step. *arXiv preprint arXiv:2411.10440*.

Yanzhi Xu, Yueying Hua, Shichen Li, and Zhongqing Wang. 2024b. Exploring Chain-of-Thought for Multi-modal Metaphor Detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 91–101.

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. 2019. Multi-Target Embodied Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6309–6318.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. 2022. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language.

Hang Zhang, Xin Li, and Lidong Bing. 2023a. Video-LLaMA: An Instruction-Tuned Audio-Visual Language Model for Video Understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023b. Multimodal Chain-of-Thought Reasoning in Language Models. *arXiv preprint arXiv:2302.00923*.

Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards Automatic Learning of Procedures From Web Instructional Videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

A Prompt Templates

A.1 Default

Instruction: An instructor is helping a user make a pinwheel. The ingredients required and the steps to complete are:

Ingredients:
1 8-inch flour tortilla
Jar of nut butter
Jar of jelly

Recipe:
Step 1: Place a tortilla on the cutting board.
Step 2: Scoop and spread nut butter onto the tortilla.
Step 3: Clean the knife with a paper towel.
Step 4: Scoop and spread jelly over the nut butter.
Step 5: Clean the knife with a paper towel.
Step 6: Roll the tortilla from one end to the other into a log shape, about 1.5 inches thick.
Step 7: Secure the rolled tortilla by inserting 5 toothpicks about 1 inch apart.
Step 8: Trim the ends of the tortilla roll.
Step 9: Slide floss under the tortilla.
Step 10: Slice the tortilla roll with the floss.
Step 11: Continue slicing with floss to create 5 pinwheels.
Step 12: Place the pinwheels on a plate.

The following is a summary of the current situation:
Instructor: oh we'll be trimming the edges with a butter knife
User: ok got it
User: do we have any scissors? oh i see them
Instructor: what would you need the scissors for?
User: place the scissors halfway between two toothpicks
Instructor: oh we'll be using dental floss to cut the pinwheel
User: OK
User: OK cut through the roll
User: done
Instructor: nice

<OPTIONAL IMAGE CAPTION>

Based on the image you see and the situation described, which step of the recipe do you think the user is currently on? Explain why.

A.2 Socratic

Turn 1: You will be presented with different pieces of evidence and your task is to predict which step the user is most likely completing based on this evidence. Initially, all steps are equally likely but you will need to update these probabilities when given evidence. You don't need to give explicit probabilities, just one of either very low, low, high, or very high. Also assume that the steps are usually executed in order. Given this dialogue history update the step probabilities accordingly. <*DIALOGUE HISTORY*>

Turn 2: Which step is the user currently performing? Give an initial guess and explain why.

Example initial guess: Step 2. The evidence provided indicates that the user has the tortilla and the nut butter. These ingredients are specifically required for Step 2, which involves spreading the nut butter onto the tortilla.

Turn 3: Given your current guess, if the step was successfully completed, what would you expect to see?

Turn 4: Now, consider this image of the current scene as evidence. Is the user actively interacting with any objects in the scene with their hands? If so, do these align with your previous expectations? If the image is irrelevant or unhelpful to your deduction, say so. Update the step probabilities accordingly. <*SCENE IMAGE*>

Turn 5: Therefore, combine this information with the dialogue history provided and give a final guess for which step the user is currently performing. Provide the final guess only in your response.

Example final guess: Step 5.

A.3 Socratic w/ Classifier

Turn 1: You will be presented with different pieces of evidence and your task is to predict which step the user is most likely completing based on this evidence. Initially, the step probabilities are:
Step 1: Low
Step 2: Low
Step 3: Very High
Step 4: Low
Step 5: Low
Step 6: Low
Step 7: Low
Step 8: Low
Step 9: Low
Step 10: Low
Step 11: Low
Step 12: Low

The first piece of evidence is this dialogue history between the instructor and user. Update the step probabilities accordingly. <*DIALOGUE HISTORY*>

Continue as in standard Socratic setup.

B Hand-selected Images

B.1 Pinwheel

(a) **Step 2**: Scoop and spread nut butter onto the tortilla.

(b) **Step 8**: Trim the ends of the tortilla roll.

(c) **Step 12**: Place the pinwheels on a plate.

Figure 4: Hand-selected images for Steps 2, 8 and 12 for the pinwheel task.

B.2 Coffee

(a) **Step 1**: Measure 12 ounces of cold water and transfer to a kettle.

(b) **Step 3**: Place the filter cone in the dripper.

(c) **Step 8**: Drain the coffee into the mug.

Figure 5: Hand-selected images for Steps 1, 3 and 8 for the coffee task.

B.3 Cake

(a) **Step 4**: Add oil, water, and vanilla to the mixing bowl.

(b) **Step 8**: Check that the cake is done with a toothpick.

(c) **Step 12**: Apply the frosting around the base of the cake.

Figure 6: Hand-selected images for Steps 4, 8 and 12 for the cake task.

C Training Hyperparameters

Hyperparameter	Value
Learning rate	1e-4
Batch size	32
Epochs	10
LoRA r	8
LoRA α	8

Table 4: Training hyperparameters for the ViT step classifier models.