MMPlanner: Zero-Shot Multimodal Procedural Planning with Chain-of-Thought Object State Reasoning

Afrina Tabassum
Amazon
afrinat@amazon.edu

Bin Guo
Alexa, Amazon
guobg@amazon.com

Xiyao Ma
Alexa, Amazon
maxiya@amazon.com

Hoda Eldardiry
Virginia Tech
hdardiry@vt.edu

Ismini Lourentzou
University of Illinois Urbana-Champaign
lourent2@illinois.edu

Abstract

Multimodal Procedural Planning (MPP) aims to generate step-by-step instructions that combine text and images, with the central challenge of preserving object-state consistency across modalities while producing informative plans. Existing approaches often leverage large language models (LLMs) to refine textual steps; however, visual object-state alignment and systematic evaluation are largely underexplored. We present **MMPlanner**, a zero-shot MPP framework that introduces Object State Reasoning Chain-of-Thought (OSR-CoT) prompting to explicitly model object-state transitions and generate accurate multimodal plans. To assess plan quality, we design LLM-as-a-judge protocols for planning accuracy and cross-modal alignment, and further propose a visual step-reordering task to measure temporal coherence. Experiments on RecipePlan and WikiPlan show that MMPlanner achieves state-of-the-art performance, improving textual planning by $+6.8\%$, cross-modal alignment by $+11.9\%$, and visual step ordering by $+26.7\%$.
https://plan-lab.github.io/mmplanner

1 Introduction

Procedural planning involves generating a sequence of steps to accomplish a goal (Lyu et al., 2021), *e.g.*, *baking a cake* or *assembling a bookshelf*. Domains such as robotics (Kovalchuk et al., 2021; Zhao et al., 2023), reasoning systems (Chen et al., 2017; Wei et al., 2022), *etc*., rely on effective procedural planning. Consequently, the field has received growing attention, driven by the recent advancements in LLMs (Liu et al., 2023; Zhu et al., 2023). Existing works utilize task-specific concept knowledge (Sun et al., 2023), knowledge from LLMs (Yuan et al., 2023), or multimodal input (Zhou et al., 2023; Wang et al., 2023), and generate linear (Wang et al., 2023; Yuan et al., 2023; Sun et al., 2023) or non-linear (Zhou et al., 2023) textual procedural plans. However, text-only instructions often lack the visual clarity and specificity required for complex tasks, limiting understanding, accessibility, and engagement. Multimodal Procedural Planning (MPP) addresses these limitations by jointly generating textual step instructions with corresponding step images, yielding more precise and accessible procedural knowledge.

A central challenge in MPP is generating step visuals that accurately reflect object state transitions. These transitions can be explicit, when the change is clearly described in the current textual step, or implicit, when it must be inferred from prior steps or broader context. For instance, in Figure 1, step 2 explicitly describes mixing ingredients, so the corresponding image should depict a bowl containing the mixed dry ingredients. Step 3, in contrast, involves adding butter, and the image must implicitly incorporate the existing mixture from Step 2. In this case, the visual should show butter being added into the bowl of mixed dry ingredients, even though ingredients are not restated in the step text.

Another important challenge involves the evaluation of multimodal plans, particularly in determining whether the generated steps successfully accomplish the intended task. Prior work has primarily measured semantic similarity between generated and reference textual plans (Wang et al., 2023; Yuan et al., 2023), which cannot effectively verify true task completion. Furthermore, text-based semantic metrics overlook critical dimensions such as visual–text alignment, temporal coherence, and the informativeness of visual steps. As a result, evaluation of multimodal plans still relies heavily on human judgment (Lu et al., 2024; Wang et al., 2023), which is labor-intensive, difficult to scale, and prone to inconsistency across annotators.

To address these challenges, we introduce **M**ulti**M**odal **Planner** (**MMPlanner**), a zero-shot framework for generating consistent multimodal plans that capture both explicit and implicit ob-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 18623–18639
November 4-9, 2025

Figure 1: **Multimodal Procedural Planning.** Left: **MMPlanner**, processes overall goals to produce comprehensive step-by-step textual and visual plans. Right: Our proposed evaluation assesses the planning accuracy of the textual plan, the cross-modal alignment between visual and textual steps, and the temporal coherence of the visual steps.

ject state changes across visual steps. MMPlanner leverages Chain-of-Thought (CoT) prompting with **O**bject **S**tate **R**easoning (**OSR-CoT**) to guide the model in reasoning about object transitions across steps. To the best of our knowledge, MMPlanner is the first MPP approach to generate zero-shot multimodal plans that jointly model implicit and explicit state changes through prompted reasoning, without requiring task-specific training.

To enable scalable and automatic evaluation of MPP, we introduce a set of customized multimodal LLM-based evaluators and propose a comprehensive evaluation framework comprising: (1) **T**extual-**Plan Score** (**T-PlanScore**), which measures the planning accuracy of the generated textual plan by assessing the alignment between the goal and the generated textual steps; (2) **C**ross-modal **A**lignment **Score** (**CA-Score**), which evaluates the relevance between generated step images and corresponding textual steps; and (3) **V**isual **S**tep **Ordering** (**VS-Ordering**), a task that assesses the informativeness and temporal coherence of the visual plan by recovering the correct step order from shuffled images. The contributions of our work are:

- We introduce **MMPlanner**, a zero-shot MPP method that generates coherent multimodal plans reflecting object state changes in visual plan sequences. We empirically validate MMPlanner on two benchmark datasets, achieving improvements of up to 6.8% in textual plan quality, 11.9% in cross-modal alignment, and 26.7% in visual step ordering accuracy.
- MMPlanner incorporates background context from previous task steps through an **Object State Reasoning Chain-of-Thought (OSR-CoT)** prompting strategy, enabling explicit modeling of evolving object states across steps, and reducing inference time by $\sim$46.25% compared to SoTA MPP baselines.
- We propose a **reference-free evaluation** framework to assess planning accuracy, cross-modal alignment, temporal coherence, and visual informativeness of generated plans, achieving stronger correlation with human judgments than prior cross-modal metrics ($\rho = 0.57$ vs. 0.37 for CLIPScore) while reducing step-level evaluation time by $\sim$66% (90s $\rightarrow$ 30s). For textual planning, our automated evaluation requires only $\sim$0.7s per plan compared to $\sim$5mins for human assessment, enabling large-scale evaluation.

2 Related Work

Procedural Planning. Procedural planning methods fall into two categories: selection-based and generation-based. Selection-based approaches (Zhao et al., 2023; Lu et al., 2022; Song et al., 2023; Wu et al., 2022; Zhou et al., 2022; Ashutosh et al., 2023) rely on predefined candidates, limiting generalization to unseen scenarios. Generation-based methods, powered by LLMs (Zhu et al., 2023; Ouyang et al., 2022), focus on generating textual plans (Wang et al., 2023; Sun et al., 2023). Recently, TIP (Lu et al., 2024) generates multimodal plans by prompting an LLM and image generation model twice sequentially, increasing the inference time. In summary, existing methods often fail to accurately reflect changes in object states throughout the steps.

Although recent work explores tracking state changes in videos (Niu et al., 2024) and leverages large-scale datasets (Souček et al., 2025), these approaches typically assume access to full video sequences (Niu et al., 2024) and struggle to maintain state consistency across frames (Souček et al.,

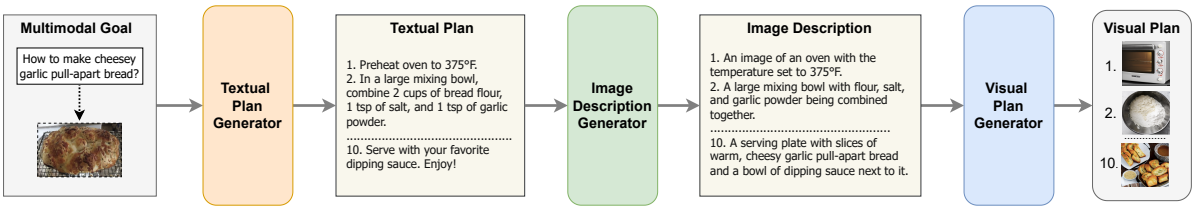

Figure 2: **MMPlanner Overview.** Given a goal instruction, MMPlanner first generates a corresponding visual goal. Then, the Textual Plan Generator produces textual plans aligned with this multimodal goal. Each step is passed to the Image Description Generator, which produces detailed visual descriptions capturing explicit and implicit object state changes. Finally, the Visual Plan Generator uses these descriptions to create step-by-step images, resulting in a comprehensive multimodal plan that maintains consistency across both textual and visual steps.

2025). In addition, Statler (Yoneda et al., 2024) focuses on maintaining object states for embodied robotic reasoning with low-level, fine-grained actions in closed environments. In contrast, our work generates visual plans from scratch in a zero-shot setting, given only a goal, and emphasizes high-level, interpretable steps aimed at human-centric multimodal procedural planning.

Beyond procedural planning, prior work has explored cross-modal coherence and multimodal discourse, focusing on temporal and narrative consistency across modalities (Alikhani et al., 2019; Inan et al., 2021). Such studies show that modeling discourse relations and coherence cues can strengthen alignment and narrative flow. Our work is complementary, and future extensions could incorporate discourse-aware prompting to further enhance temporal and narrative consistency in visual plans.

Multimodal Plan Evaluation. Procedural plans can be evaluated manually or automatically. Human evaluations (*e.g.*, crowdsourcing) can be time-intensive and error-prone, while automatic metrics such as WMD (Kusner et al., 2015), Sentence-BERT (Reimers and Gurevych, 2019), *etc*., though scalable, fall short in assessing temporal relationships and completeness of textual plans. For evaluating multimodal plans, prior work measures similarity between textual plans and captions from visual plans, encountering similar limitations (Lu et al., 2024). To the best of our knowledge, there is currently a lack of automatic frameworks for evaluating the quality of multimodal plans. To address this gap, we propose T-PlanScore for task completion and CA-Score for cross-modal alignment evaluation. Additionally, we introduce a visual reordering task to assess temporal coherence.

3 Method

The goal of MPP is to generate a sequence of steps, each with a textual and visual component, that together achieve a high-level task goal. Given a high-level goal instruction $\mathcal{G}_t$ that outlines the task, the objective is to generate a sequence of low-level steps $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, where n denotes the number of steps. Each step s_i comprises a textual description t_i and a corresponding step image v_i, denoted as (t_i, v_i). The step-wise textual and visual plans are denoted as $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, respectively.

3.1 MMPlanner Overview

We introduce MMPlanner, a method for generating a multimodal plan $\mathcal{S} = (\mathcal{T}, \mathcal{V})$ from a given task goal $\mathcal{G}_t$, consisting of (1) a Textual Plan Generator that produces a sequence of textual steps $\mathcal{T}$ from goal $\mathcal{G}_t$, (2) an Image Description Generator that produces detailed image descriptions $\mathcal{D}$ from textual steps $\mathcal{T}$, capturing both explicit and implicit object state changes, and (3) a Visual Plan Generator that generates visual plans $\mathcal{V}$ from descriptions $\mathcal{D}$. An overview is shown in Figure 2.

3.2 Textual Plan Generator

Recent advancements in LLMs (Taori et al., 2023) have facilitated the generation of step-by-step textual plans from a high-level goal (Lu et al., 2024; Wang et al., 2023; Sun et al., 2023; Yuan et al., 2023). However, text-only goal instructions $\mathcal{G}_t$ often under-specify the task and omit visual cues such as object appearances or spatial configurations that are crucial for accurate plan generation. These missing cues can lead to ambiguous or incomplete plans, especially in tasks requiring implicit state reasoning. To address this, we enhance task comprehension by extending goal instructions $\mathcal{G}_t$ to include a corresponding visual goal $\mathcal{G}_v$, generated using Stable Diffusion (Rombach et al., 2022). The resulting multimodal goal is denoted as $\mathcal{G} = (\mathcal{G}_t, \mathcal{G}_v)$, where $\mathcal{G}_t$ and $\mathcal{G}_v$ represent textual and visual goals, respectively. We then utilize a

VLM (Liu et al., 2023) to generate a step-by-step textual plan $\mathcal{T}$ from $\mathcal{G}$.

3.3 Image Description Generator

While Text-to-Image (T2I) generation models can produce images from the information explicitly present in the textual descriptions (Rombach et al., 2022), they struggle to interpret implicit state changes from textual steps. For example, when prompted with the step instructions "*3. Add 1 cup of unsalted butter, cut into small pieces, and mix until the mixture is crumbly*", Stable Diffusion produces an image of butter, unable to infer the concept of a "mixture" due to the lack of explicit information present in the prompt. To address this limitation, following recent works (Niu et al., 2024; Menon and Vondrick, 2022), we employ an LLM (Brown et al., 2020) to generate image descriptions from textual plans, leveraging inherent commonsense knowledge. Specifically, we feed the LLM with each textual step along with the overall goal and previous steps to offer additional context and enable it to infer implicit object state changes. A corresponding simple prompt for generating image descriptions is as follows:

```
Prompt:
In the process of [goal], current step is [step].
The previous steps are [prev_steps]. Describe an
image containing the items involved in the
current step, after completing the current step.
Focus on the items and their physical states.
Answer: <Image Description>
```

However, with this prompt, the model neglects key details like texture and often struggles to contextualize previous steps as background information, thereby tending to include extraneous details from previous steps in the image description of the current step. For instance, for the input [step] illustrated in Figure 3, the above prompt generates:

"*The image shows a bowl of bread flour mixture with small pieces of unsalted butter. The flour is white and powdery, while the salt and garlic powder are both fine grains*".

Here, the model overlooks the texture "crumbly mixture" and hallucinates irrelevant details about flour, salt, and garlic powder from previous steps.

Chain-of-Thought Prompting with Object State Reasoning (OSR-CoT): To address these challenges, inspired by (Niu et al., 2024), we introduce OSR-CoT, a Chain-of-Thought (CoT) prompting strategy designed to reduce hallucinations by guiding the model through a stepwise reasoning process. First, OSR-CoT prompts the model to describe the current step in detail, incorporating relevant background information from the overall goal and previous steps. Next, OSR-CoT instructs the model to reason about object state changes before and after the current step. Finally, it directs the model to incorporate these state changes into a coherent image description. As illustrated in Figure 3, OSR-CoT generates concise image descriptions, including both explicit and implicit object state changes. Unlike the previously introduced simple prompt, the image description generated by OSR-CoT (Figure 3) is concise without unnecessary hallucinated information. We denote the sequence of generated image descriptions as $\mathcal{D} = \{d_1, \ldots, d_i, \ldots, d_n\}$ where n is the total number of steps and d_i is the image description for the i-th step.

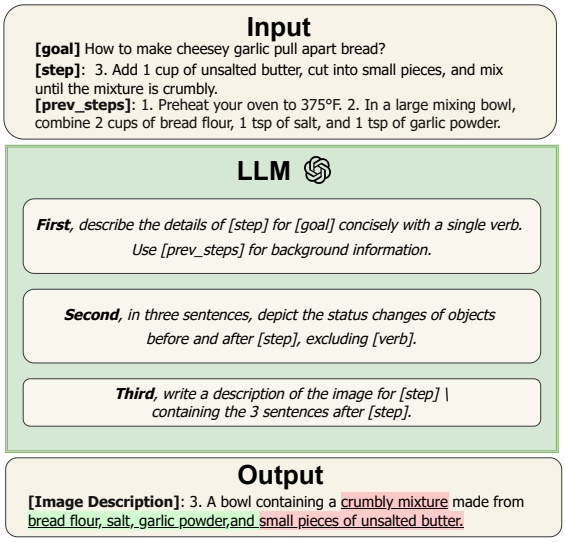

Figure 3: **Image Description Generator**. Our proposed OSR-CoT prompts the LLM to include both explicit (red) and implicit (green) state changes of the associated objects in the generated image description.

3.4 Visual Plan Generation

Given a step image description d_i, we generate a corresponding step image v_i using Stable Diffusion (Rombach et al., 2022). Since Stable Diffusion is a stochastic generative model, prompting it multiple times with the same description d_i yields a set of diverse images $\mathcal{I}_i = \{I_{i1}, \ldots, I_{ik}, \ldots, I_{iK}\}$. Empirically, we observe that while some samples accurately reflect the fine-grained attributes in d_i, others may miss key visual details. As shown in Figure 4, only I_2 captures both the "crumbly" texture and the "mixture" mentioned in the description, whereas I_1 and I_K fail to depict these elements. To ensure consistency and visual fidelity across steps, we sample multiple candidates and select the one that best aligns with the textual description.

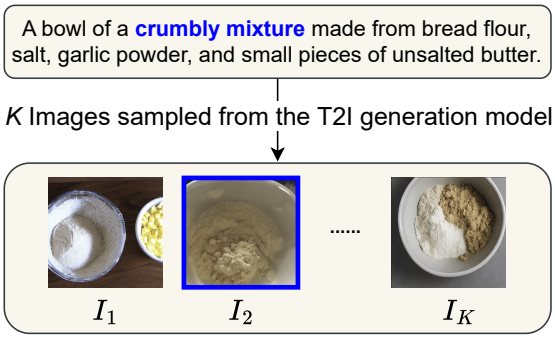

Figure 4: **Cross-modal Step Image Selection** showing multiple images generated with SD given the same image description as prompt for step "*3. Add 1 cup of unsalted butter, cut into small pieces, and mix until the mixture is crumbly*". Here, I_2 successfully captures "*crumbly*" and "*mixture*" textures (blue bbox), whereas I_1 and I_K fail to incorporate these fine-grained details.

We introduce a cross-modal selection strategy to choose the best image from $\mathcal{I}_i$. Each image sample is assigned a similarity score based on its alignment with the description d_i in the feature space. To map d_i and $\mathcal{I}_i$ into a shared feature space, we utilize a pretrained BLIP-2 (Li et al., 2023) feature extractor. In BLIP-2, a Querying Transformer (Q-Former) is trained to bridge the gap between a frozen image encoder and an LLM, where the objective is to generate a visual feature representation that is relevant to the prompt and interpretable by the LLM. Let f_{d_i} and f_{ik} denote the BLIP-2 feature embeddings of the image description d_i and the k-th sampled image I_{ik}, respectively, where $1 \leq k \leq K$. We select an image based on cross-modal similarity $\arg\max_k sim(f_{ik}, f_{d_i})$, where $sim(\cdot, \cdot)$ refers to cosine similarity. This process is repeated for all steps to obtain the final visual plan $\mathcal{V}$ consisting of all generated step images.

4 MPP Evaluation

LLMs have demonstrated strong performance on complex reasoning tasks, motivating their use as automated evaluators that often surpass human workers in efficiency (Gilardi et al., 2023). Building on this insight, we introduce CA-Score, which measures the alignment between each textual step and its corresponding visual depiction, and T-PlanScore, which evaluates whether a generated textual plan is both task-consistent and logically coherent. In practice, T-PlanScore completes evaluation in an average of 0.7 seconds per task, compared to approximately 5 minutes for human annotators. For complex tasks (*e.g.*, "*How to weave a rag rug?*"), human evaluation takes a longer time due to domain-specific knowledge requirements. Similarly, for CA-Score, each step-level evaluation completes in about 1 minute, representing a 66% reduction in time compared to human assessment, which averages 3 minutes. These results highlight the efficiency and practicality of LLM-based evaluation for multimodal planning tasks.

T-PlanScore. Prior approaches to textual plan evaluation often rely on semantic similarity to reference plans (Lu et al., 2024), which may not fully capture planning accuracy or temporal coherence. We propose T-PlanScore, a reference-free method that prompts a language model (Brown et al., 2020) to assess how well a generated plan aligns with the overall task goal. The prompt guides the model to consider both procedural correctness and logical step ordering. Empirically, we find that higher T-PlanScore values correspond to more coherent plans that accurately reflect the intended procedure.

CA-Score. Cross-modal alignment between textual and visual plans is often evaluated using similarity-based metrics such as CLIPScore (Hessel et al., 2021) or Sentence-BERT (Reimers and Gurevych, 2019), computed between generated visual captions and reference textual plans (Lu et al., 2024). While effective for coarse semantic matching, these methods may fail to capture finer-grained alignment, such as implicit object state changes not explicitly described in the text. To address this, we leverage multimodal language models, which have shown strong cross-modal reasoning capabilities (Zhu et al., 2023; Liu et al., 2023). Inspired by recent work on text-to-image evaluation (Huang et al., 2024), we employ a VLM with Chain-of-Thought (CoT) to assess cross-modal alignment at the step level. Specifically, we prompt the model to describe the contents of a generated image and compare the resulting description with the corresponding step text, evaluating alignment in terms of depicted actions and object states. We refer to the resulting score as CA-Score.

VS-Ordering. To assess the informativeness and temporal coherence of visual plans, we additionally introduce a visual step reordering task. Given an unordered sequence of visual steps, the objective is to recover their correct temporal order. This task provides a direct measure of how well the visual outputs capture procedural structure. A related task was proposed by Wu et al. (2022), who introduced multimodal instruction sequencing involving both textual and visual inputs. However, we find that including text in the sequence biases

Model	RecipePlan				WikiPlan			
	T-PlanScore ↑	S-BERT ↑	WMD ↑	METEOR ↑	T-PlanScore ↑	S-BERT ↑	WMD ↑	METEOR ↑
Text-Ref+SD	73.85	0.72	0.13	0.09	62.53	**0.78**	0.70	0.20
GPT-3.5+SD	80.43	0.73	0.86	0.14	81.93	0.75	**0.77**	0.09
TIP	82.00	0.73	0.86	0.14	83.00	**0.78**	**0.77**	0.09
MMPlanner	**82.05**	**0.78**	**0.88**	**0.15**	**84.43**	0.77	0.75	**0.23**

Table 1: **Textual Evaluation on RecipePlan and WikiPlan.** Across both datasets, MMPlanner consistently surpasses or achieves competitive performance against baselines.

evaluation toward the textual modality, limiting sensitivity to visual quality (Appendix D). To address this, our formulation focuses solely on the visual modality, enabling the evaluation of visual procedural understanding independent of textual cues. We adopt a pretrained visual sequencing model (Wu et al., 2022) that consists of a CLIP image encoder and an order decoder based on the BERSON framework (Cui et al., 2018). The vision encoder is trained with self-supervised objectives such as masked language modeling, patch-based image swapping, and sequential masked region modeling. For each visual plan, we randomly shuffle the step order and use the model to predict the correct sequence. Figure 1 illustrates an example.

5 Experiments

We evaluate MMPlanner on the RecipePlan and WikiPlan (Lu et al., 2024) datasets. RecipePlan consists of 1,000 recipe tasks adapted from RecipeQA (Yagcioglu et al., 2018), where each task includes a goal (taken from the recipe title) and a sequence of text-image pairs representing procedural steps. WikiPlan contains 1,000 tasks sourced from WikiHow articles, where the article title serves as the goal, the main body text forms the textual plan, and accompanying images comprise the visual plan. We conduct experiments comparing **MMPlanner** with **TIP** (Lu et al., 2024), a dual prompting MPP method that integrates procedural knowledge from LLMs and T2I models by prompting both twice during inference. We also compare against two baselines from TIP: (1) **GPT-3.5+SD**, which independently generates textual plans using GPT-3.5 and visual plans using Stable Diffusion (SD); and (2) **Text-Ref+SD**, which generates images with Stable Diffusion (SD) from brief step titles instead of detailed steps.

Our evaluation is structured across three dimensions: (i) **textual planning**, which assesses the accuracy and coherence of the generated textual plan; (ii) **cross-modal alignment**, which evaluates the relevance between each visual step and its corresponding text; and (iii) **visual ordering**, which measures the temporal consistency and informativeness of the visual plan. Implementation details can be found in Appendix A.

Model	RecipePlan		WikiPlan	
	CA-Score ↑	CLIPScore ↑	CA-Score ↑	CLIPScore ↑
Text-Ref+SD	70.81	60.68	61.61	65.42
GPT-3.5+SD	71.49	73.00	63.18	71.08
TIP	67.68	73.09	63.30	72.17
MMPlanner	**77.07**	**77.44**	**69.23**	**76.10**

Table 2: **Cross-Modal Step-level Evaluation on RecipePlan and WikiPlan.** MMPlanner improves cross-modal alignment between visual and textual steps.

5.1 Quantitative Evaluation

Textual Evaluation. We employ T-PlanScore to assess planning accuracy, alongside standard text similarity metrics Sentence-BERT (S-BERT) (Reimers and Gurevych, 2019), Word Movers Distance (WMD) (Kusner et al., 2015), and METEOR (Banerjee and Lavie, 2005). Table 1 compares MMPlanner against baselines. Overall, MMPlanner achieves strong performance across both datasets. On T-PlanScore, GPT-3.5+SD, TIP, and MMPlanner perform similarly, reflecting the effectiveness of LLMs in producing coherent goal-aligned textual plans. In contrast, Text-Ref+SD performs worse due to the limited information available in step titles used as input. Unlike reference-based metrics, T-PlanScore does not rely on a fixed ground-truth sequence. Instead, it provides a reference-free assessment of how well the generated plan aligns with the task goal, accommodating multiple valid solution paths. On feature similarity metrics (S-BERT, WMD, METEOR), MMPlanner consistently outperforms baselines, particularly on RecipePlan, indicating strong semantic alignment with the reference plans.

Cross-modal Step-Level Evaluation. We evaluate step-level cross-modal alignment using CLIPScore (Hessel et al., 2021) and our proposed CA-Score. As shown in Table 2, MMPlanner out-

Model	RecipePlan						WikiPlan					
	Acc ↑	LCS ↑	τ ↑	Dist. ↓	MS ↓	WMS ↓	Acc ↑	LCS ↑	τ ↑	Dist. ↓	MS ↓	WMS ↓
Text-Ref+SD	22.60	2.09	0.04	7.76	2.66	6.07	19.70	2.80	0.01	7.87	2.78	6.43
GPT-3.5+SD	21.65	1.83	0.03	7.78	2.63	5.99	18.95	2.69	0.01	7.66	2.67	6.23
TIP	21.70	1.81	0.05	7.79	2.66	5.99	18.79	2.83	0.02	7.92	2.75	6.30
MMPlanner	**27.50**	**3.09**	**0.22**	**6.51**	**2.39**	**4.99**	**23.43**	**2.91**	**0.05**	**7.60**	**2.60**	**5.90**

Table 3: **Visual Sequence Ordering (VS-Ordering) Evaluation.** MMPlanner consistently outperforms baselines.

performs all baselines on both datasets. On CA-Score, MMPlanner improves over TIP by 11.9% on RecipePlan and 9.37% on WikiPlan. Additionally, our Cross-modal Step Image Selector yields CLIPScore improvements of 6.1% and 5.5% over TIP on RecipePlan and WikiPlan, respectively. These results demonstrate that MMPlanner produces visual plans that are more semantically aligned with their corresponding textual steps. Importantly, CA-Score shows stronger correlation with human ratings ($\rho = 0.57$) compared to CLIPScore ($\rho = 0.37$) (details in Appendix D.3), underscoring the reliability of our proposed metric.

Visual Ordering. We evaluate the temporal coherence of generated visual step sequences on the VS-Ordering task with six established ordering metrics: Accuracy (Acc), Distance (Dist), Minimum Swap (MS), Weighted Minimum Swap (WMS), Longest Common Subsequence (LCS), and Kendall's Tau (τ) (Lapata, 2003). Detailed metric definitions can be found in Appendix B. Table 3 reports results on both datasets. On RecipePlan, MMPlanner outperforms all methods by substantial margins, achieving gains of 26.7%, 16.4%, 10.15%, and 16.7% over the second-best method (TIP) on Accuracy, Dist, MS, and WMS, respectively. On WikiPlan, MMPlanner shows consistent improvements over TIP with relative gains of 24.7%, 4.0%, 5.5%, and 6.3% on the same metrics. On RecipePlan, MMPlanner surpasses TIP by 47.85% in LCS and by over 340% in Kendall's Tau, further indicating stronger global temporal consistency in the generated visual plans.

Inference Comparison. TIP requires two sequential prompts for text-to-image and image-to-text models, resulting in increased inference time. In contrast, MMPlanner integrates reasoning over previous steps and object states directly via OSR-CoT, eliminating the need for dual prompting and significantly streamlining inference. As a result, MMPlanner achieves an average inference time of 52.02 seconds, compared to 96.77 seconds for TIP — a relative reduction of approximately 46.25%.

Model	CA-Score ↑	CLIPScore ↑	Acc ↑	Dist.↓	MS ↓	WMS ↓
RecipePlan						
LLaVa+SD	72.02	73.87	22.10	7.71	2.70	6.41
+ OSR-CoT	75.15	75.12	25.50	7.03	2.53	5.16
+ Previous Steps	75.27	75.19	26.80	6.82	2.48	5.03
+ CM Sel. (MMPlanner)	**77.07**	**77.55**	**27.50**	**6.51**	**2.39**	**4.99**
WikiPlan						
LLaVa+SD	71.65	72.83	22.01	7.70	2.62	6.06
+ OSR-CoT	74.58	74.27	22.29	7.66	2.61	5.98
+ Previous Steps	75.70	74.35	23.17	7.78	2.60	6.16
+ CM Sel. (MMPlanner)	**77.48**	**76.10**	**23.43**	**7.60**	**2.60**	**5.90**

Table 4: **Ablation on MMPlanner Components.** MMPlanner's components collectively improve cross-modal alignment and temporal coherence.

5.2 Ablation Studies

We conduct ablation studies to analyze the contributions of MMPlanner components (Section 5.2 and Appendix C). We evaluate: (1) the impact of each MMPlanner module; (2) the importance of different components within the OSR-CoT prompt; (3) the effectiveness of BLIP-2 as a cross-modal feature extractor; (4) the influence of the sampling hyperparameter K in visual selection; and (5) the role of the visual goal in textual plan generation. We further validate our proposed evaluation by: (1) assessing the reliability of T-PlanScore; (2) evaluating the correlation between CA-Score and human judgments; and (3) analyzing robustness across different LLMs/VLMs (Appendix D).

MMPlanner Components. We conduct an ablation to evaluate the contribution of each module in MMPlanner. We begin with a baseline variant, **LLaVa+SD**, where textual plans are generated using LLaVa and directly passed to Stable Diffusion (SD) to produce visual steps. We then integrate the Image Description Generator component with Object State Reasoning via Chain-of-Thought prompting (**OSR-CoT**), without conditioning on previous steps. Next, we incorporate prior step context into OSR-CoT (**Previous Steps**), followed by the integration of our Cross-modal Step Image Selection module (**CM Sel.**). As shown in Table 4, each component incrementally improves performance, with the full model (MMPlanner) achieving the highest scores in both cross-modal alignment and VS-Ordering. Notably, the introduction of OSR-

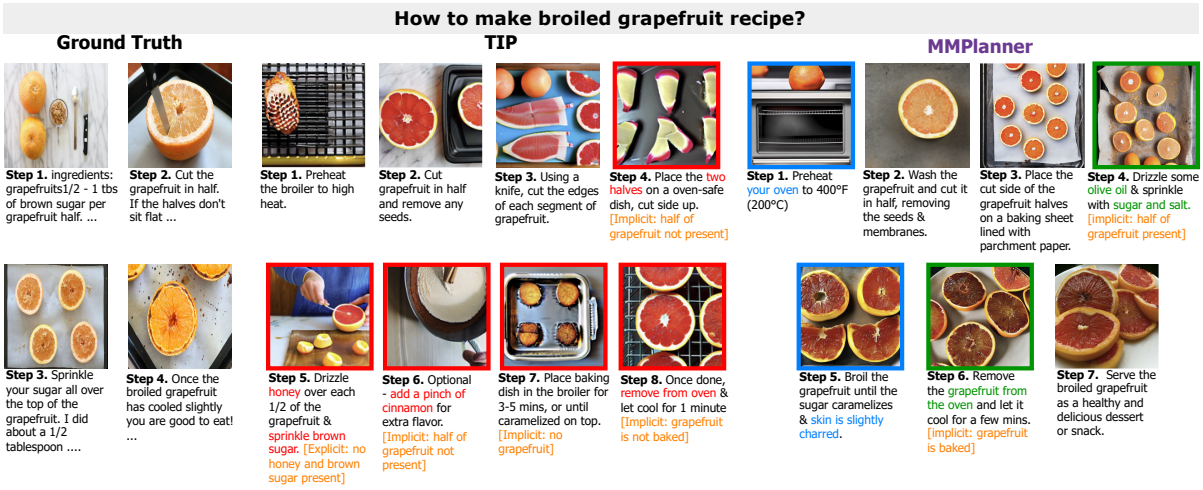

Figure 5: **Qualitative Comparison TIP vs. MMPlanner on RECIPEPLAN task "How to make broiled grapefruit recipe".** Explicit state changes are those clearly described in the textual step and expected in the visual step. Implicit state changes are not explicitly stated in the step text but are necessary to convey in the visual step. Left: TIP fails to accurately reflect both explicit and implicit object state changes in visual steps (red text and bboxes). Right: MMPlanner captures both explicit (blue) and implicit (green) object state changes. Explanations in orange text.

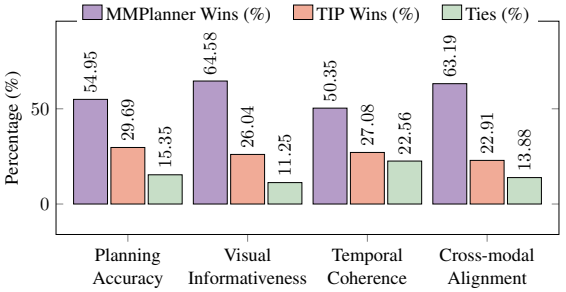

Figure 6: **User Study MMPlanner vs. TIP.** Multimodal plans evaluated by participants across four dimensions. Bars present percentage of wins and ties.

CoT leads to a substantial gain in CA-Score on both RECIPEPLAN and WIKIPLAN, underscoring the value of structured image descriptions. While improvements on VS-Ordering are more modest for WIKIPLAN, the inclusion of OSR-CoT, previous step context, and CM Sel. still results in consistent gains across Acc., Dist., and WMS.

5.3 Human Evaluation

We conduct a user study to compare MMPlanner and TIP across four key dimensions: **1) Planning Accuracy**, *i.e.*, whether following the multimodal plan would successfully complete the task; **2) Visual Informativeness**, *i.e.*, how well the visual steps support task execution; **3) Temporal Coherence**, *i.e.*, whether the steps are presented in a logical order; and **4) Cross-modal Alignment**, *i.e.*, the consistency between each step image and its corresponding textual step. We conducted a human evaluation with 26 participants who assessed multimodal plans generated by TIP and MMPlanner across 12 distinct tasks. Each participant compared plans from both models across four key dimensions, resulting in a total of 1,248 pairwise judgments. For each task, participants were presented with two unlabeled step-by-step multimodal plans, one from each model, alongside the high-level task objective and were asked to choose their preferred plan based on four criteria: (i) accuracy of the steps in achieving the task goal; (ii) visual informativeness of each step; (iii) temporal coherence across steps; and (iv) alignment between textual and visual modalities. This setup ensures an unbiased and comprehensive evaluation of plan quality. As shown in Figure 6, MMPlanner receives consistently higher preference across all evaluation criteria, demonstrating improved planning accuracy, visual clarity, temporal structure, and visual-text consistency compared to TIP.

5.4 Qualitative Evaluation

We compare multimodal plans generated by TIP and MMPlanner. Figure 5 presents example step-by-step plans for the task "*How to make broiled grapefruit recipe*". TIP's visual steps often fail to capture key object state changes, both explicit and implicit, as described in the textual instructions. This observation aligns with its lower CA-Score scores. In contrast, MMPlanner generates visual steps that more accurately reflect the described explicit (highlighted with blue color) and implicit (highlighted with green color) object states. Furthermore, the visual plan produced by MMPlan-

ner is qualitatively closer to the ground truth plan, demonstrating better alignment between text and image modalities and stronger procedural understanding. Additional examples in Appendix E.

6 Broader Impacts

This work aims to advance the deployment of multimodal generative models, such as LLMs and text-to-image models, for real-world, step-by-step task assistance. Our goal is to make task-driven assistive technology more practical and accessible, particularly for users who benefit from visual guidance. We acknowledge the limitations of generative models, especially their susceptibility to hallucination and misinformation. To address this, OSR-CoT encourages grounded reasoning by decomposing tasks into smaller, verifiable steps, reducing the risk of unsupported outputs. Future directions could focus on integrating external knowledge verification to further enhance the reliability and trustworthiness of AI-generated multimodal plans.

7 Limitations

MMPlanner leverages LLMs and VLMs for multimodal plan generation and evaluation via T-PlanScore and CA-Score. However, hallucinations remain a known limitation of LLMs (Xu et al., 2024). This issue is most evident before applying OSR-CoT (Section 3.3). For example, the generated description of the task "*How to make cheese garlic pull-apart bread?*" (Figure 7) for step 5 is "*A bowl of dough mixture is forming. Flour and butter can be seen in the background*" without OSR-CoT, showing that the LLM introduces unrelated details about flour and butter from earlier steps. In contrast, OSR-CoT yields "*The milk mixture being slowly stirred into the dry ingredients*", more accurately aligning with the step's intent and focusing on what is relevant (ingredients, processes, *etc*) for that specific step. This shows how OSR-CoT reduces hallucinations and improves step relevance. OSR-CoT improves object state reasoning, but MMPlanner does not explicitly enforce visual consistency for peripheral elements such as cookingware shape or material. For instance, in Figure 7, the bowl depicted in steps 2, 3, and 5 varies in appearance. MMPlanner inherits limitations from Stable Diffusion, particularly its inability to render concepts absent from its training data. For example, given the step "*Beat the egg with a fork*" from the task "*How to make an omelet*", the model fails to generate an accurate depiction of a "beaten egg". Addressing such inconsistencies remains an avenue for future work. Finally, while T-PlanScore shows consistent monotonic trends under plan degradations, its absolute calibration can be imperfect. In some cases, low-quality plans (*e.g.*with many deleted steps) still receive high scores. This reflects a broader limitation of LLM-based evaluators, where prompt adherence and human alignment are not guaranteed. Future work could improve calibration through human feedback, refined prompts, or preference-based fine-tuning. As multimodal LLMs continue to improve, they offer potential for better MPP evaluation frameworks, but further work is needed to refine both generation and evaluation.

8 Conclusion

We present MMPlanner, a zero-shot multimodal procedural planning method using OSR-CoT prompting to capture explicit and implicit object state changes. To evaluate generated plans, we propose an automatic evaluation that assesses planning accuracy, cross-modal alignment, and temporal coherence. Experiments show MMPlanner generates accurate and coherent multimodal plans.

9 Acknowledgments

This research is based on work partially supported by the Amazon–Virginia Tech Initiative for Efficient and Robust Machine Learning and by the U.S. Defense Advanced Research Projects Agency (DARPA) under award numbers HR00112390062 and HR001125C0303. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Amazon, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. 2019. CITE: A corpus of image-text discourse relations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL)*.

Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. 2023. Video-mined task graphs for keystep recognition in instructional videos. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

F. Gilardi, M. Alizadeh, and M. Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *National Academy of Sciences*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

K. Huang et al. 2024. T2I-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mert Inan, Piyush Sharma, Baber Khalid, Radu Soricut, Matthew Stone, and Malihe Alikhani. 2021. Cosmic: A coherence-aware generation metric for image descriptions. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Alexander Kovalchuk, Shashank Shekhar, and Ronen I Brafman. 2021. Verifying plans and scripts for robotics tasks using performance level profiles. In *International Conference on Automated Planning and Scheduling*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning (ICML)*.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Association for Computational Linguistics (ACL)*.

J. Li et al. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. In *NeurIPS Workshop on Instruction Tuning and Instruction Following*.

Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2022. Neuro-symbolic procedural planning with commonsense prompting. In *International Conference on Learning Representations (ICLR)*.

Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Wang, and William Yang Wang. 2024. Multimodal procedural planning via dual text-image prompting. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Qing Lyu, Li Zhang, and Callison-Burch Chris. 2021. Goal-oriented script construction. In *International Conference on Natural Language Generation*.

Sachit Menon and Carl Vondrick. 2022. Visual classification via description from large language models. In *International Conference on Learning Representations (ICLR)*.

Yulei Niu, Wenliang Guo, Long Chen, Xudong Lin, and Shih-Fu Chang. 2024. Schema: State changes matter for procedure planning in instructional videos. In *International Conference on learning Representations (ICLR)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Philip Sedgwick. 2014. Spearman's rank correlation coefficient. *BMJ: British Medical Journal*, 349.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tomáš Souček, Prajwal Gatti, Michael Wray, Ivan Laptev, Dima Damen, and Josef Sivic. 2025. Showhowto: Generating scene-conditioned step-by-step visual instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chenkai Sun, Tie Xu, Cheng Xiang Zhai, and Heng Ji. 2023. Incorporating task-specific concept knowledge into script learning. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Qingyun Wang, Manling Li, Hou Pong Chan, Lifu Huang, Julia Hockenmaier, Girish Chowdhary, and Heng Ji. 2023. Multimedia generative script learning for task planning. In *Association for Computational Linguistics (ACL)*.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2022. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. In *Association for Computational Linguistics (ACL)*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv:2401.11817*.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R. Walter. 2024. Statler: State-maintaining language models for embodied reasoning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Jankowski, Yanghua Xiao, and Deqing Yang. 2023. Distilling script knowledge from large language models for constrained language planning. In *Association for Computational Linguistics (ACL)*.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. In *RSS Workshop on Learning for Task and Motion Planning*.

Shuyan Zhou et al. 2022. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Association for Computational Linguistics (ACL)*.

Yu Zhou, Sha Li, et al. 2023. Non-sequential graph script induction via multimedia grounding. In *Association for Computational Linguistics (ACL)*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations (ICLR)*.

A Implementation Details

We employ LLaVa-1.5-7B (Liu et al., 2023) and GPT-3.5 (Brown et al., 2020) for the Textual Plan Generator (Section 3.2) and Image Description Generator (Section 3.3), respectively. For T-PlanScore and CA-Score, we utilize GPT-3.5 (Brown et al., 2020) and MiniGPT-4 (Zhu et al., 2023), respectively. Prompts are described below.

Textual Plan Generator: Given a multimodal goal, we construct a prompt that asks the VLM to generate a textual step-by-step plan, *i.e.*,

```
Using the image as a reference, and goal
"How to make garlic pull-apart bread?",
give step-by-step brief instructions,
according to the following format:
1. Start each step with the step number.
2. 1 sentence of 50 words maximum for each step.
```

Image Description Generator. We introduce Chain-of-Thought with Object State Reasoning (OSR-CoT) prompting designed to generate descriptions for visual steps based on textual instructions. To manage token limitations in GPT-3.5, we cap the number of prior steps ([prev_steps]) used as background information to 10, closely matching the average number of ground truth steps (8.92). The prompt also includes an in-context example to guide the model's reasoning. For example, the **OSR-CoT prompt** and **in-context example** for the **[goal]** "How to make cheesy garlic pull-apart bread" and example **[step]** "3. Add 1 cup of unsalted butter, cut into small pieces, and mix until the mixture is crumbly" is as follows:

```
First, describe details of [step] for [goal]
with one verb. Use [prev_steps] for
background information.
Second, use 3 sentences to describe the state
changes of objects before and after [step],
avoiding using [verb].
Third, write description of the [step] image
containing the 3 sentences after [step].
[goal]: Task: How to make fried egg with cheese.
[step]: 3. Pour a small amount of butter
        or oil into the pan.
[prev_steps]: 1. Crack an egg into a bowl.
              2. Heat a non-stick frying pan
              on medium heat.

Description:
Pour a small amount of butter or oil into a pan.
Before:
- An egg is cracked in a bowl.
- Non-stick frying pan is heated on medium heat.
- The pan is empty without any butter/oil in it.
After:
-The butter or oil is in the pan.
- The pan is coated with butter or oil.
- The pan is ready for cooking the egg.
Image Description:
A non-stick frying pan with butter or oil
poured into it.

[goal]: How to make a cheesy garlic pull-apart
        bread?
[step]: 3. Add 1 cup of unsalted butter, cut
        into small pieces, and mix until the
        mixture is crumbly.
[prev_steps]: 1. Preheat oven to 375°F (190°C).
              2. In a large mixing bowl, combine
              2 cups of bread flour, 1 tsp salt,
              and 1 tsp of garlic powder.
Description: ....
```

T-PlanScore. We utilize the following prompt template for computing the T-PlanScore:

```
You are my assistant in evaluating the alignment
between the overall goal [goal] and the
step-by-step instructions [steps].
[goal]: How to make tomato chutney?
[steps]: 1. Gather Ingredients ..........
Evaluate how well [goal] aligns with [steps]
Give a score from 0 to 100, according to
the following criteria:
100:[steps] perfectly describe the steps for
    completing [goal].
80: [steps] mostly describe the steps for
    completing [goal] but with minor
    discrepancies in the step ordering.
60: [steps] describe the steps for completing
    [goal], but missed some important steps.
40: [steps] didn't describe steps for completing
    [goal] as it has discrepancies in step
    ordering and missed few important steps.
20: [steps] completely failed to describe the
    steps for completing the [goal] as it has
    lots of discrepancies in step ordering and
    missed a lot of important steps.
Provide your analysis and explanation in JSON
format with the following keys:
score, explanation.
```

The LLM returns a JSON-formatted output with "score" and "explanation" as keys.

CA-Score: For CA-Score, we prompt VLM with two questions sequentially. First, we prompt the model to describe the contents of the visual step.

```
You are my assistant to evaluate the corres-
pondence of the image to a given text prompt.
Briefly describe the image within 50 words.
Focus on the objects in the image and their
attributes, such as color, shape, texture, and
action relationships.
```

Then, based on the answer, we ask the model to assign a step image-text cross-modal alignment score using the following prompt:

```
According to the image and your previous answer,
evaluate how well the image describes action in
step: [step], a subprocess of the task [goal].
Give a score from 0 to 100, according to
the following criteria:
100: the image perfectly describes the action of
     [step] and object states after [step],
     with no discrepancies.
80:  the image portrayed most of the action of
     [step] and object states after [step],
     but with minor discrepancies.
60:  the image depicted some action of [step]
     and object states after [step] but ignored
     some key parts or details.
40:  the image did not depict any action of
     [step] and object states after [step].
20:  the image failed to convey the full action
     of [step] and object states after [step].
Provide your analysis and explanation in JSON
format with the following keys:
score and explanation.
```

B Evaluation Metrics

Cross-modal Step-level Evaluation. We utilize CLIPScore (Hessel et al., 2021), *i.e.*CLIP embedding similarity between visual and textual steps, and our proposed CA-Score that accounts for alignment in object states and implied actions. CLIPScore was originally designed for image captioning and hence may underperform in MPP settings where visual steps contain implicit cues or elements not explicitly stated in the corresponding text. We report average CLIPScore and CA-Score across all steps and tasks. Both scores are normalized to a 1–100 scale, with higher values indicating stronger cross-modal alignment.

Textual Evaluation. In addition to T-PlanScore, which evaluates planning accuracy and temporal coherence, we report traditional text generation metrics: **Sentence-BERT (S-BERT)** (Reimers and Gurevych, 2019), **Word Movers Distance (WMD)** (Kusner et al., 2015), and **METEOR** (Banerjee and Lavie, 2005). S-BERT and WMD measure feature-level similarity, and METEOR captures word-level lexical similarity between generated and reference text plans. Following TIP, we compute WMD-based similarity over sentence embeddings, where higher values denote stronger alignment. All metrics are reference-based and normalized to [0,1], whereas T-PlanScore produces reference-free scores in [0,100].

VS-Ordering. VS-Ordering evaluates predicted step order with position-based metrics. **Accuracy (Acc)** is the percentage of steps in the correct absolute position (range 0–100), and **Distance (Dist)** is the average positional deviation (Dist $\geq$ 0). **Longest Common Subsequence (LCS)** measures the average overlap in subsequences (0 to sequence length), while **Kendall's Tau (τ)** (Lapata, 2003) quantifies pairwise order consistency via $\tau = 1 - \frac{2 * \#inversion}{\#pairs}$, where $\#inversion$ is the number of pairs in the predicted order with incorrect relative order, and $\#pairs = \binom{n}{2}$, with τ ranging from –1 to 1. **Minimum Swap (MS)** is the minimum number of swaps needed to recover the correct order (0 to sequence length–1), and **Weighted Minimum Swap (WMS)** penalizes larger swap distances (non-negative, unbounded). Higher Acc, LCS, and τ indicate stronger ordering, while lower Dist, MS, and WMS indicate fewer deviations. Following Wu et al. (2022), we evaluate on the first five sequence steps.

C Ablations on MMPlanner Components

C.1 OSR-CoT Prompt Ablation

OSR-CoT consists of three key components: (1) a one-shot example illustrating the image description generation process (**1-Shot**), (2) reasoning about the current step (**Desc.**), and (3) reasoning about object state changes before and after the step (**State**). To assess the contribution of each component, we conduct an ablation study with three variants, where components are added incrementally. In OSR-CoT-V1, the LLM is prompted to generate an image description using only the [goal], [step], and [prev_steps] without utilizing any of these components. OSR-CoT-V2 adds the one-shot example to guide the model with a concrete reference. The detailed prompt for OSR-CoT-V2 is as follows:

```
Write description of the [step] image
containing the 3 sentences after [step].
Use [prev_steps] for background information.
[goal]: Task: How to make fried egg with cheese.
[step]: 3. Pour a small amount of butter
        or oil into the pan.
[prev_steps]: 1. Crack an egg into a bowl.
              2. Heat a non-stick frying pan
              on medium heat.
Image Description: A non-stick frying pan with
butter or oil poured into it.

[goal]: How to make a cheesy garlic pull-apart
        bread?
[step]: 3. Add 1 cup of unsalted butter, cut
        into small pieces, and mix until the
        mixture is crumbly.
[prev_steps]: 1. Preheat oven to 375°F (190°C).
              2. In a large mixing bowl, combine
              2 cups of bread flour, 1 tsp salt,
              and 1 tsp of garlic powder.
Image Description: ....
```

OSR-CoT-V3 extends V2 by incorporating the description component, prompting the model to

Dataset	Model	1-Shot	Desc.	State	CA-Score ↑	CLIPScore ↑	Acc ↑	LCS ↑	τ ↑	Dist.↓	MS ↓	WMS ↓
RECIPE PLAN	OSR-CoT-V1	✗	✗	✗	71.85	75.11	24.42	2.88	0.07	7.59	2.67	5.87
	OSR-CoT-V2	✓	✗	✗	73.42	75.62	24.49	2.90	0.14	6.99	2.68	5.59
	OSR-CoT-V3	✓	✓	✗	74.33	77.18	26.21	3.01	0.16	6.82	2.48	5.13
	OSR-CoT (MMPlanner)	✓	✓	✓	**77.07**	**77.55**	**27.50**	**3.09**	**0.22**	**6.51**	**2.39**	**4.99**
WIKI PLAN	OSR-CoT-V1	✗	✗	✗	70.31	72.98	20.10	2.80	0.02	7.90	2.70	6.30
	OSR-CoT-V2	✓	✗	✗	72.11	74.67	23.17	**2.98**	**0.05**	7.67	2.66	5.97
	OSR-CoT-V3	✓	✓	✗	75.42	75.59	**23.72**	2.82	0.04	7.71	2.69	**5.87**
	OSR-CoT (MMPlanner)	✓	✓	✓	**77.44**	**76.10**	23.43	2.91	**0.05**	**7.60**	**2.60**	5.90

Table 5: **Ablation on OSR-CoT Components.** Incrementally adding each component improves cross-modal alignment and temporal coherence in the generated visual plans.

[goal]: How to make broiled grapefruit recipe?
[step]: Drizzle some olive oil & sprinkle with sugar and salt.
[prev_steps]: 1. Preheat your oven to 400°F (200°C). 2. Wash the grapefruit and cut it in half, removing the seeds & membranes. 3. Place the cut side of the grapefruit halves on a baking sheet lined with parchment paper.
Description:
Some olive oil is drizzled and sprinkled with brown sugar and a pinch of salt over the grapefruit halves.
Before:
- The oven is preheated to 400°F (200°C).
- The grapefruit is cut in half, with seeds and membranes removed.
- The cut side of the grapefruit halves are placed on a baking sheet lined with parchment paper.
After:
- Olive oil is drizzled over the grapefruit halves.
- Brown sugar is sprinkled over grapefruit halves.
- A pinch of salt is added to each half.
Image Description:
The image shows two halved grapefruits on a baking sheet, one side coated with olive oil, brown sugar, and a pinch of salt.

[goal]: How to make cheese garlic pull-apart bread?
[step]: In a separate bowl, combine 1 cup of milk, 1 egg, and half cup of grated cheddar cheese.
[prev_steps]: 1. Preheat your oven to 375°F. 2. In a large mixing bowl, combine 2 cups bread flour, 1 tsp salt, and 1 tsp garlic powder. 3. Add 1 cup butter, cut into small pieces, and mix until mixture is crumbly.
Description:
1 cup of milk, 1 egg, and 1/2 cup of grated cheddar cheese is combined in a separate bowl.
Before:
- The oven is preheated to 375°F (190°C).
- In a large mixing bowl, there is bread flour, salt, and garlic powder mixed together.
- There are small pieces of unsalted butter added to the mixture.
After:
- Ingredients in the separate bowl are combined into one mixture.
- The mixture contains milk, egg, and grated cheddar cheese.
- It's ready to be poured over prepared bread dough before baking.
Image Description:
A bowl containing milk, egg, and grated cheddar cheese are being mixed together.

Table 6: **Qualitative Examples** of generated descriptions by prompting with the proposed OSR-CoT method.

first describe the current step in detail before generating the corresponding image description.

```
First, describe details of [step] for [goal]
with one verb. Use [prev_steps] for
background information.
Second, write description of the [step] image
containing the 3 sentences after [step].
[goal]: Task: How to make fried egg with cheese.
[step]: 3. Pour a small amount of butter
        or oil into the pan.
[prev_steps]: 1. Crack an egg into a bowl.
              2. Heat a non-stick frying pan
              on medium heat.
Description:
Pour a small amount of butter or oil into a pan.
Image Description:
A non-stick frying pan with butter or oil
poured into it.
[goal]: How to make a cheesy garlic pull-apart
        bread?
[step]: 3. Add 1 cup of unsalted butter, cut
        into small pieces, and mix until the
        mixture is crumbly.
[prev_steps]: 1. Preheat oven to 375°F (190°C).
              2. In a large mixing bowl, combine
              2 cups of bread flour, 1 tsp salt,
              and 1 tsp of garlic powder.
Description: ....
```

Finally, the full OSR-CoT prompt incorporates the state-change component, guiding the model to reason about object transitions before and after each step. The complete prompt is provided in Appendix A. As shown in Table 5, both CA-Score and CLIPScore improve steadily with the inclusion of each component, underscoring their collective role in producing accurate, state-aware image descriptions. Table 6 presents qualitative examples.

Dataset	FE	CA-Score ↑	CLIPScore ↑	τ ↑	Dist.↓	MS ↓	WMS ↓
RECIPE PLAN	No FE	75.27	75.19	0.16	6.82	2.48	5.03
	CLIP	75.85	75.67	0.14	7.01	2.46	5.15
	BLIP-2	**77.07**	**77.55**	**0.22**	**6.51**	**2.39**	**4.99**
WIKI PLAN	No FE	75.70	74.35	0.03	7.78	2.60	6.16
	CLIP	75.76	76.00	0.04	7.67	2.64	5.97
	BLIP-2	**77.44**	**76.10**	**0.05**	**7.60**	**2.60**	**5.90**

Table 7: **Ablation on Cross-Modal Feature Extractors (FEs)** with no feature extractor (No FE), CLIP, and BLIP-2 for cross-modal step image selection.

C.2 Cross-modal Feature Extractor

Effectiveness of BLIP-2 as Feature Extractor. To evaluate the impact of different cross-modal feature extractors (FEs) in step image selection (Section 3.4), we compare: (1) **No FE**, which selects an image based solely on the step image description without any kind of cross-modal feature extraction; (2) **CLIP**, a retrieval-based model; and (3) **BLIP-2**, the cross-modal feature extractor used in MMPlanner. As shown in Table 7, BLIP-2 consistently outperforms other variants, likely due to its alignment strategies, which better capture fine-grained visual-textual correspondence.

Ablation on Hyperparameter K. We investigate the effect of the hyperparameter K in the cross-modal step image selector, which determines the number of candidate images generated per step. Specifically, we vary K across the following values: $K \in \{1, 5, 10, 15, 20\}$. As shown in Table 8, increasing K leads to consistent gains in CA-Score

Dataset	K	CA-Score ↑	CLIPScore ↑	τ ↑	Dist.↓	MS ↓	WMS ↓
RECIPE PLAN	1	75.27	75.19	0.16	6.82	2.48	5.02
	5	75.34	75.30	0.19	6.76	2.44	**4.93**
	10	75.43	75.27	0.20	6.68	2.41	4.97
	15	76.05	77.24	0.20	6.64	2.41	5.00
	20	**77.07**	**77.55**	**0.22**	**6.51**	**2.39**	4.99
WIKI PLAN	1	75.70	74.35	0.03	7.78	2.61	6.16
	5	75.94	75.14	0.03	7.75	2.63	6.16
	10	75.88	75.21	0.04	7.68	2.64	6.13
	15	76.11	76.12	0.04	7.71	2.61	6.13
	20	**77.44**	**76.10**	**0.05**	**7.60**	**2.60**	**5.90**

Table 8: **Ablation on Cross-Modal Step Image Selection Hyperparameter *K*** (number of generated images).

Dataset	$\mathcal{G}_v$	T-PlanScore ↑	S-BERT ↑	WMD ↑	METEOR ↑
RECIPE PLAN	✗	80.88	0.75	0.76	0.10
	✓	**82.05**	**0.78**	**0.88**	**0.15**
WIKI PLAN	✗	82.30	**0.77**	**0.76**	0.20
	✓	**84.43**	**0.77**	0.75	**0.23**

Table 9: **Ablation on Multimodal Goal.** Generating textual plans w/ and w/o a Goal Image.

and CLIPScore, indicating improved alignment between selected images and step text. While improvements in VS-Ordering metrics are modest, the results suggest that higher K values enhance visual relevance and semantic fidelity w.r.t. the corresponding step texts.

C.3 Ablation on Multimodal Goals

We conduct an ablation to analyze the role of goal image $\mathcal{G}_v$ in generating textual plans. As shown in Table 9, incorporating the visual goal consistently improves performance compared to using the textual goal alone, demonstrating that the goal image $\mathcal{G}_v$ contributes complementary information that enhances the quality of generated textual plans.

D Ablations on MPP Evaluation

D.1 Robustness of T-PlanScore and CA-Score

We evaluate T-PlanScore and CA-Score across different LLMs and VLMs. As shown in Table 10, MMPlanner consistently achieves the highest scores across all configurations, demonstrating its effectiveness independent of the underlying evaluation model. Moreover, the consistent trends across baselines confirm that both T-PlanScore and CA-Score serve as stable and reliable metrics for evaluating procedural plans.

D.2 T-PlanScore Reliability

Both RECIPEPLAN and WIKIPLAN include tasks requiring domain expertise, such as "*How to fix a leaky faucet*" and "*How to pasteurize*". Given the complexity of these tasks, human evaluation for assessing plan accuracy would be costly and labor-intensive. Instead, to evaluate the reliability of T-PlanScore, we conduct an ablation study by perturbing LLM-generated plans using two strategies: (i) *random permutation*, which shuffles the step order; and (ii) *random deletion*, which randomly removes 50% of the textual plan steps. We compute T-PlanScore using both GPT-3.5 and LLaVa-1.5-13B to assess its robustness across model types. As shown in Table 11, T-PlanScore consistently degrades when steps are deleted or permuted, demonstrating its sensitivity to structural disruptions in the plan. In contrast, standard metrics such as WMD, METEOR, and S-BERT exhibit minimal variation and fail to capture these structural inconsistencies. This highlights T-PlanScore's unique ability to penalize violations in temporal coherence, which are often overlooked by traditional text similarity metrics. Furthermore, we vary the deletion percentage to test granularity. Table 12 shows T-PlanScore increases with more complete plans, demonstrating its sensitivity to missing steps.

D.3 CA-Score and Human Correlation

To evaluate how well CA-Score aligns with human judgment, we conducted a human study involving 30 step-image examples: 10 from ground truth, 10 from TIP, and 10 from MMPlanner. 14 annotators independently assessed each image's relevance to its paired textual instruction, resulting in 420 human ratings in total (30 examples × 14 raters). Evaluations were performed on a 5-point Likert scale, with annotators instructed to consider both the depicted action and object states, as well as the image's alignment with the overarching task goal. A score of 1 reflects an image that is irrelevant to both the step and the overall goal, whereas a score of 5 signifies perfect alignment with both. To evaluate inter-annotator agreement, we compute both the weighted Cohen's kappa and Spearman's rank correlation coefficient (Sedgwick, 2014), obtaining scores of 0.61 and 0.67, respectively, indicating moderately strong inter-rater agreement.

Table 13 presents the average human rating along with CA-Score and CLIPScore for the collected examples, suggesting that raters preferred the step images generated from MMPlanner while step images generated by TIP are perceived to be less accurate or relevant compared to the ground truth and MMPlanner generated plans. Finally, to assess the alignment between automated metrics and human evaluation, we compute Spearman's

Model	CA-Score				T-PlanScore			
	RecipePlan		WikiPlan		RecipePlan		WikiPlan	
	MiniGPT-4	LLaVa	MiniGPT-4	LLaVa	GPT-3.5	LLaMa	GPT-3.5	LLaMa
Text-Ref + SD	71.47	55.65	70.81	54.51	73.85	71.66	62.53	55.20
GPT-3.5 + SD	72.24	56.42	71.49	55.52	80.43	85.86	81.93	85.36
TIP	70.08	54.49	67.68	55.76	82.00	87.12	83.00	90.43
MMPlanner	**77.07**	**63.10**	**77.44**	**62.59**	**82.05**	**87.67**	**84.43**	**90.46**

Table 10: **Ablation on LLM/VLMs** used in LM-based evaluation CA-Score and T-PlanScore.

Permute	Delete	RecipePlan					WikiPlan				
		T-PlanScore (GPT-3.5)	T-PlanScore (LLaMa)	WMD	METEOR	S-BERT	T-PlanScore (GPT-3.5)	T-PlanScore (LLaMa)	WMD	METEOR	S-BERT
✓	✓	77.50	55.8	0.87	0.09	0.76	78.10	66.34	0.74	0.19	**0.77**
✗	✓	78.12	60.68	0.87	0.09	0.76	80.10	70.75	0.74	0.19	0.76
✓	✗	78.60	64.78	**0.89**	**0.15**	**0.77**	80.20	74.64	**0.75**	**0.23**	**0.77**
✗	✗	**82.05**	**79.81**	0.88	**0.15**	**0.77**	**84.43**	**86.07**	**0.75**	**0.23**	**0.77**

Table 11: **Verifying T-PlanScore** on MMPlanner's textual plans with unordered or missing steps.

Deletion %	RecipePlan		WikiPlan	
	GPT-3.5	LLaMa	GPT-3.5	LLaMa
80	75.02	50.56	77.14	68.01
60	76.88	61.23	79.26	75.04
50	78.12	70.13	80.10	77.18
40	78.38	76.02	80.43	80.70
20	80.51	80.10	81.57	82.55
0	**82.05**	**86.67**	**84.43**	**89.46**

Table 12: **T-PlanScore Ablation** with varying % of missing steps for plans generated by MMPlanner.

rho (ρ) for both CA-Score and CLIPScore against human ratings, yielding a correlation of 0.57 for CA-Score and 0.37 for CLIPScore, suggesting CA-Score reflects human judgment better.

D.4 Motivation of VS-Ordering Task

Wu et al. (2022) propose a multimodal sequencing task that assesses temporal coherence by predicting the correct order of an unordered multimodal plan (text and image steps). We apply this task to evaluate the output of baseline models under the hypothesis that more interpretable and expressive plans would yield higher reordering accuracy. However, as shown in Table 14, all baselines perform similarly, largely due to their accurate textual plans. Since the sequencing model primarily relies on textual cues, improvements in visual quality have a limited impact. To better isolate visual coherence, we instead adopt the vision-only reordering model from Wu et al. (2022), where gains in visual planning directly enhance task performance.

E Qualitative Examples

Figure 7 compares TIP and MMPlanner on the task "*How to make cheese garlic pull-apart bread?*". In this example, TIP generates a generic cheese

Model	CA-Score ↑	CLIPScore ↑	Human Rating ↑
GroundTruth	86.00	77.16	4.19
TIP	58.00	76.00	2.51
MMPlanner	**85.00**	**79.19**	**4.19**

Table 13: **Comparison of CA-Score, CLIPScore, and Human Ratings** for step image-text pairs evaluated by humans across ground truth (GroundTruth), TIP, and MMPlanner generated plans.

Modality	Model	Acc ↑	LCS ↑	τ ↑	Dist.↓	MS ↓	WMS ↓
Multi modal	TIP	74.20	4.34	0.80	1.80	0.75	1.02
	MMPlanner	**74.43**	**4.37**	**0.80**	**1.70**	**0.73**	**1.01**
Vision Only	TIP	21.70	1.81	0.05	7.79	2.66	5.99
	MMPlanner	**27.50**	**3.09**	**0.22**	**6.51**	**2.39**	**4.99**

Table 14: **Comparison of Multimodal and Vision-Only Sequence Ordering on RecipePlan.**

block for step 7, failing to reflect the dish-specific context (*pull-apart bread*), which is not explicitly mentioned in the text. In contrast, MMPlanner correctly depicts the baked bread in step 8, despite the step only referring to baking the dough, demonstrating its ability to infer object state transitions beyond surface text. Figures 8 and 9 provide additional qualitative examples on two WikiPlan tasks, "*How to Get a Sick Kitten to Eat?*" and "*How to Weave a Rag Rug?*", respectively. In Figure 8, while all TIP-generated images include a kitten, they lack consistency across steps (*e.g.*, step 5) and often omit explicit objects mentioned in the text, such as the dish/bowl in step 1 and the food in step 4. MMPlanner produces step images that more faithfully reflect the textual instructions and maintain higher visual consistency. In Figure 9, TIP fails to infer implicit object state information such as "rag rug" in step 5. MMPlanner, however, includes the rug in steps 5-6, demonstrating its ability to maintain visual consistency across step images.

Figure 7: **Qualitative Comparison between TIP and MMPlanner on RECIPEPLAN for the task goal "How to make cheese garlic pull-apart bread".** Left: In step 7, TIP fails to incorporate the dough state in the generated step image, as it was not mentioned in the textual step (orange text). Moreover, in step 6, TIP does not depict the "shredded cheese" in the step image, which is explicitly mentioned in the textual step (red text and bboxes). Right: In step 9, MMPlanner depicts the correct state of "baked loaf" (green) even if it was not mentioned in the textual step (orange text). In step 3, the generated step image illustrates the explicit object state "crumbly" (blue).

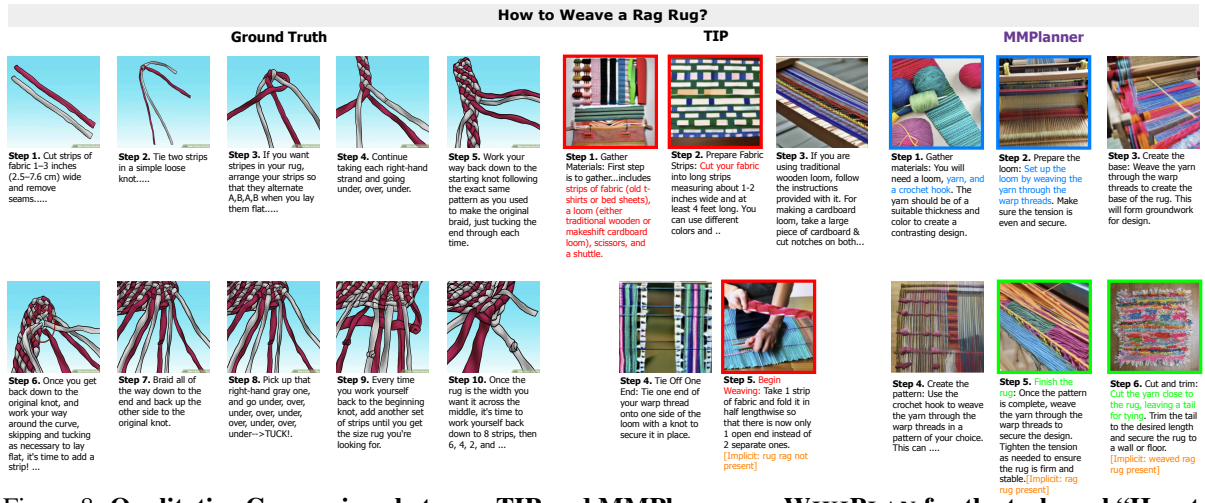

Figure 8: **Qualitative Comparison between TIP and MMPlanner on WIKIPLAN for the task goal "How to Get a Sick Kitten to Eat".** Left: In step 1, TIP fails to incorporate the explicit object states in the generated step image (red texts and boxes). Right: In step 1, MMPlanner incorporates the explicit state of the foods (blue).

Figure 9: **Qualitative Comparison between TIP and MMPlanner on WIKIPLAN for the task goal "How to weave rag rug".** Left: In step 1, TIP fails to incorporate the explicit object states in the generated step image (red texts and boxes). Moreover, in step 5, the step image does not contain the implicit object information "rag rug" (orange text). Right: In step 1, MMPlanner incorporates the explicit state of the ingredients (blue). In step 6, the generated step image includes the implicit object state "finished rag rug" (green).