Visualizing Entity States in Recipes by Generating Step Images

Anonymous ACL submission

Abstract

Procedural texts, such as recipes and instruction manuals, are crucial for understanding processes involving multiple entities over time. Entity state tracking, which monitors the states of specific entities at each time step, is a key task in this domain. However, existing benchmarks heavily rely on manually annotated datasets, limiting scalability. We propose a novel task of step image generation in recipes, using step images as visual supervision for tracking entity states in procedural text without relying on manually annotated data. By generating step images, we can visualize the entity states in each step. For this task, we collect high-quality multimodal recipe datasets, theSpruceEats. Addressing the limitation of existing two-stage methods in achieving deep interaction between text and image, this paper introduces an explicit state modeling approach based on multimodal generative models. Experiments on theSpruceEats dataset demonstrate that our method enhance entity state tracking and image generation quality compared to existing methods, improving the CLIP similarity metric by 10.2% compared to existing methods.

1 Introduction

Procedural texts, which describe processes involving one or more entities over time, such as recipes and instruction manuals, are widely spread and useful. The key task for procedural text comprehension is entity state tracking, which aims to monitor specific entities' states at each time step, described by multiple attributes such as existence or location. Popular datasets like ProPara(Dalvi et al., 2018) and RECIPE(Bosselut et al., 2018) provide benchmarks for this task. However, their reliance on manual annotations limits scalability due to substantial human resource requirements.

To address this, we propose to leverage the rich recipe data available online, particularly those with step images, to explore procedural text understanding without annotated data. Recipes, with their structured format comprising titles, descriptions, ingredient lists, and steps, serve as an excellent resource. Step images visually depict the entity states described in the text, providing visual supervision for entity state tracking.

We introduce a novel task of step image generation in recipes. As show in Figure 1, in a recipe, given textual inputs such as titles, descriptions, ingredient lists, and sequential step descriptions, our goal is to generate corresponding images for each step. By generating step images, we can visualize the entity states in each step. This task requires tracking the entities' states throughout the process to accurately generate images. To facilitate this, we collected a high-quality multimodal recipe dataset, theSpruceEats, containing 6,635 English recipes and 56,832 step images, verified by professional chefs and with consistent image quality.

The task of step image generation involves the generation of interleaved text and image. Current approaches (Li et al., 2023) typically adopt a two-stage method involving language models for generating image captions, followed by image generation models like Stable Diffusion (Rombach et al., 2022). However, these methods rely on captions as intermediaries and may not capture deep dependencies between text and images effectively.

To overcome these limitations, we utilize multimodal generative models like SEED-LLaMA (Ge et al., 2024) and LaVIT (Jin et al., 2024) to achieve unified modeling of procedural text and image, as well as the generation of step images. In these multimodal large models, images are tokenized into a sequence of image tokens, allowing them to interact and be deeply modeled alongside text tokens in large pre-trained language models like LLaMA (Touvron et al., 2023). During image generation, the model first generates image tokens using the large language model and then decodes these to-

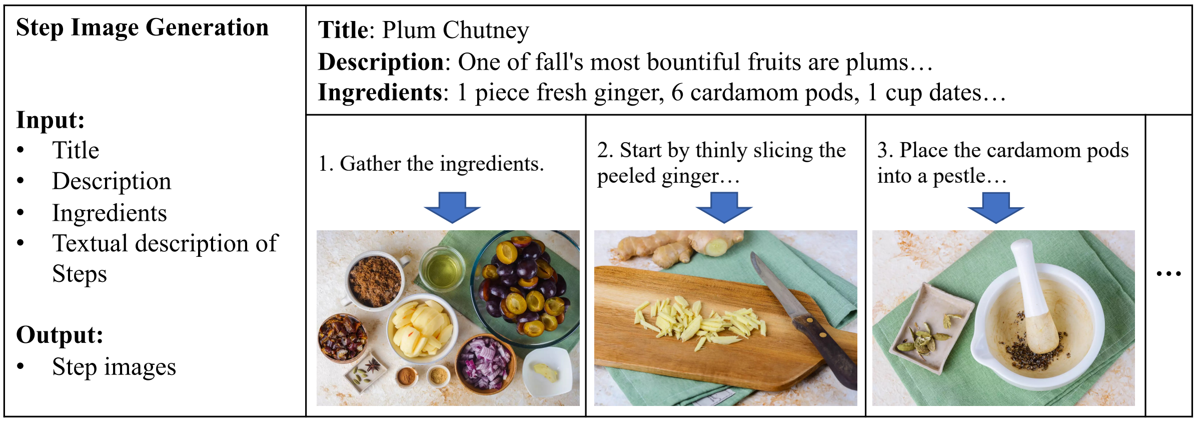

Figure 1: The definition of recipe step generation task and an example from theSpruceEats dataset.

kens into images.

For step image generation task, we first proposes two base methods: single-step image generation and step-by-step image generation. The single-step image generation method generates step images based on the textual descriptions of a recipe and the corresponding steps. The step-by-step image generation method incorporates previous steps and their step images to produce the current steps image. Moreover, we introduce an enhancement through explicit state modeling. This approach involves generating detailed textual descriptions of step images before generating corresponding image tokens. This method not only aligns more closely with pre-training tasks, thus enhancing image generation quality, but also but also breaks down complex reasoning into sequential steps similar to "chain of thought"(Wei et al., 2022), thereby reducing inference difficulty.

Experiments on theSpruceEats dataset show significant improvements on entity state tracking and image generation quality compared to existing two-stage methods, with a 10.2% increase in CLIP similarity. The multi-step image generation method outperforms the single-step method, achieving a 1.69% improvement in CLIP similarity compared to single-step methods. Furthermore, explicit state modeling enhances the quality of step image generation, with a 2.43% increase in CLIP similarity metric for single-step image generation after incorporating explicit state modeling.

Our contributions are two-fold: 1) We propose a challenging task of step image generation in recipes, leveraging step images to advance procedural text understanding, and collect a high-quality multimodal recipe dataset, theSpruceEats; 2) We propose explicit state modeling based on multi-modal generative models for this task, enhancing step image generation quality and entity state tracking.

2 Related Work

Entity state tracking is the key task in procedural text understanding. Currently, the commonly used datasets for entity state tracking tasks are ProPara and RECIPE. The former includes scientific texts that describe natural phenomena, such as the process of photosynthesis or fossil formation, with the tracking goals mainly focusing on the location and existence of entities. It contains 488 procedural texts. The latter primarily comprises cooking guide texts, where the tracking of ingredients involves attributes such as location, temperature, and composition. It includes 875 manually annotated cooking guides. The annotation of these two datasets involves a significant amount of entity state information at each step, requiring substantial human resources, making it challenging to expand the data scale.

Early methods for these datasets, such as ProGlobal(Dalvi et al., 2018), KG-MRC(Das et al., 2019), NCET(Gupta and Durrett, 2019), and IEN(Tang et al., 2020), were based on two-layer RNNs to model the step-document two-level hierarchy of procedural texts. They then obtained the state of each entity at each step by classification. Subsequent methods introduced Transformer models for procedural text modeling. For instance, REAL(Huang et al., 2021) used BERT(Devlin et al., 2019) as an encoder and employed an entity-action-location network to infer entity states. TSLM(Rajaby Faghihi and Kordjamshidi, 2021) proposed a time series language model that incorporated temporal encoding into the Transformer's input encoding to model the process. However, these methods did not consider multimodal information and could not utilize the entity state information contained in step images to aid state tracking.

3 Dataset

Existing multimodal recipe datasets include Recipe1M+(Marin et al., 2019) and RecipeQA(Yagcioglu et al., 2018). However, Recipe1M+ does not contain images of cooking steps, making it unsuitable for research on step image generation task. Although RecipeQA contains step image data, this dataset is not collected from professional recipe websites; instead, the recipes are mostly user-uploaded content, with considerable noise in both text and images, making it less suitable for learning step image generation.

To address this, we presents a high-quality multimodal recipe dataset, theSpruceEats, collected from the professional recipe website `thespruceeats.com`. This dataset includes English recipes from various regions and categories, most of which are verified by professional chefs, ensuring the quality of the recipes. The step images in this dataset mostly feature uniform backgrounds and shooting angles, clearly demonstrating the state of various entities in each step of the recipe, eliminating the interference of noise such as background, shooting angle, and watermarks, making it more suitable for learning step image generation.

This dataset contains 6,635 recipes and 56,832 step images. Some statistical data of the dataset are shown in Table 1. The theSpruceEats dataset was split into training, validation, and test sets in an 8:1:1 ratio.

Avg. Title Length	4.2 words
Avg. Description Length	23.4 words
Avg. Ingredient List Length	49.9 words
Avg. Number of Steps	8.57 steps
Avg. Step Length	20.99 words
Avg. Total Length	257.4 words

Table 1: Statistics of theSpruceEats dataset.

4 Method

To generate step images in recipes, we use multimodal generative models, which integrate text and image generation, as the base model for unified modeling and generation of procedural text and images. Different training and inference methods are proposed. Firstly, we introduces two methods: single-step image generation and step image generation. Since the task of directly generating step images from recipe text significantly differs from the pre-training tasks of the base model, an improved method based on explicit state modeling is proposed, which first generates image captions and then generates the images.

4.1 Single-Step Image Generation

In single-step image generation methods, the model generates images based on the textual descriptions of a recipe and the corresponding steps. Specifically, as shown in Figure 2, for a step s_t, the title of the recipe, description, ingredient list, textual decription of steps are concatenated together, and the following instruction is added:

> Generate an image for the step <s_t>.

as input to the model, requiring the model to generate the token sequence corresponding to the step image.

4.2 Step-by-Step Image Generation

In single-step image generation methods, all the step images are generated independently of each other, relying solely on textual information to generate the images. However, there are dependencies between the step images of different steps; entities in the images of previous steps often reappear or appear in a changed state in the images of subsequent steps. Therefore, we proposes step-by-step image generation method, where previously generated step images are incorporated as conditions when generating images for the current step.

Specifically, as shown in Figure 2, for a step s_t, the title, description, and ingredient list and text and images of steps 1 to $t-1$ are concatenated along with the following instruction:

> Generate an image for this step.

as the model's input, requiring the model to generate the token sequence corresponding to the step image. During the inference process, the image part in the input will be replaced with the step images generated in the previous steps.

4.3 Explicit State Modeling

In both single-step image generation and step image generation methods, the model implicitly models the state of entities at each step, requiring it

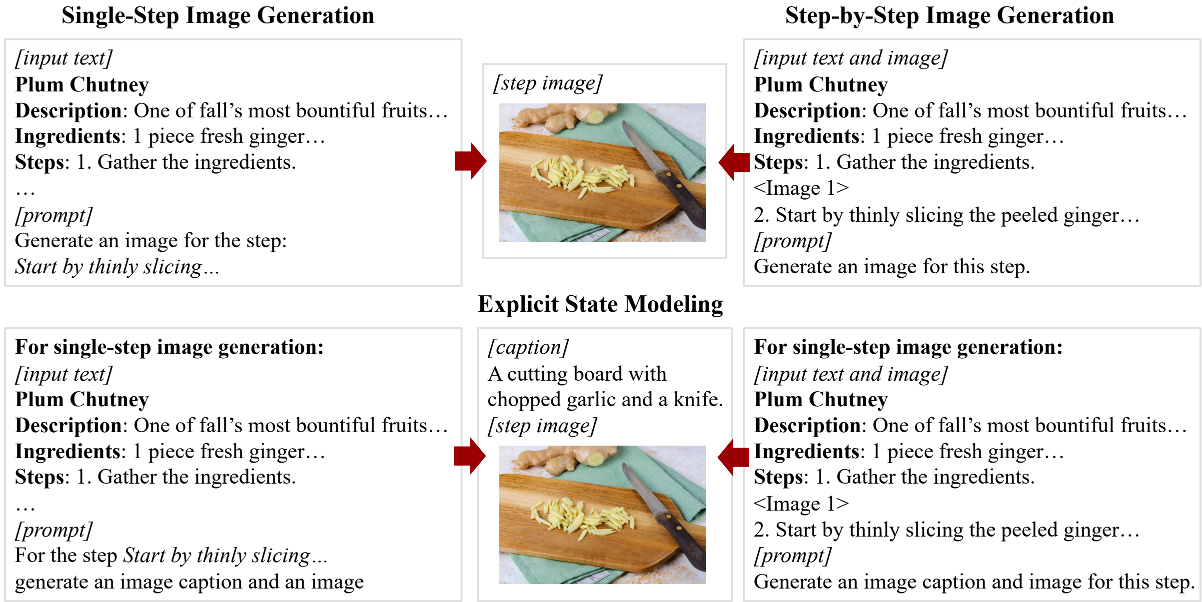

Figure 2: Illustration of proposed methods. On the top left, the single-step image generation method generates images based on the textual descriptions of a recipe and the corresponding steps. On the top right, the step-by-step image generation method incorporates previous steps and their images to produce the current step's image. At the bottom, the explicit state modeling approach enhances image generation by first generating an image caption which describes entity states before creating the final image.

to directly generate an image corresponding to the current state of the entity from the step description. However, during the model's pre-training, most of the training data is often in the form of (image caption, image) pairs, which significantly differ from the correspondence between step description and step image. To reduce the gap between the training task and the pre-training task, and inspired by the "chain of thought" method, we proposes an improved method based on explicit state modeling on the basis of the aforementioned methods.

In explicit state modeling method, we introduces the image caption of the step image as an explicit modeling of the entity states. Since most images in the theSpruceEats dataset do not contain image captions, we uses the pre-trained SEED-LLaMA model to generate image captions for the images in the dataset. For an image, the model inputs the image tokens and the prompt:

> Generate a detailed caption for this image.

then the model autoregressively generates the image caption.

After introducing the image caption of the step image, the explicit state modeling method requires the model to first generate the image caption of the step image when generating the step image, which is also an explicit modeling of the current entity states. Then, based on the input and the generated image caption, the model performs the task of generating the image.

For the single-step image generation method, in explicit state modeling method, as shown in Figure 2, for a step s_t, the title of the recipe, description, ingredient list, and operation steps are concatenated together, and the following instruction is added:

> For the step <s_t> generate an image caption and an image.

This serves as the input to the model, requiring the model to generate the token sequences of the image caption and the step image.

For the step-by-step image generation method, in the improved method based on explicit state modeling, as shown in Figure 2, for a step s_t, the title of the recipe, description, and ingredient list are concatenated together, and the text of steps 1 to $t-1$ and their corresponding step images are added, along with the prompt:

> Generate an image caption and image for this step.

This serves as the input, requiring the model to generate the token sequences of image caption and the step image. During inference, the image parts

	CLIP Sim.	FID (↓)
Baseline Method (Two-Stage)	52.40	32.55
Single-Step Image Generation (LaVIT)	61.13	28.61
+ *Explicit State Modeling*	61.62	**27.23**
Single-Step Image Generation (SEED-LLaMA)	60.20	44.54
+ *Explicit State Modeling*	**62.63**	34.70
Step-by-Step Image Generation (SEED-LLaMA)	61.60	39.96
+ *Explicit State Modeling*	62.32	31.38
Golden Image Tokens (SEED-LLaMA)	67.51	28.45
Golden Image Tokens (LaVIT)	68.53	26.55

Table 2: Experimental Results of Baseline Methods and Ours.

in the input will also be replaced by the previously generated step images.

5 Experiments

5.1 Experimental Settings

For the single-step image generation method and its improved version with explicit modeling, we selected SEED-LLaMA-8B and LaVIT-7B as the base models. For the step image generation method and its improved version with explicit modeling, we chose the pre-trained SEED-LLaMA-8B model. This is because the SEED-LLaMA model has been pre-trained on image-text interleaved datasets like MMC4(Zhu et al., 2023) and OBELICS(Laurençon et al., 2023), making it more suitable for the step image generation task. In contrast, LaVIT has not been pre-trained on similar data and may struggle to adapt to the step image generation task through fine-tuning on a smaller dataset.

For the aforementioned base model, this paper employs full-parameter fine-tuning for training. During training, the loss function is computed on the validation set, and training stops when the loss on the validation set ceases to decrease. The training hyperparameters are shown in Table 3, and each method's model training takes approximately 15 hours on 4 A40 GPUs.

When generating images, the model first generates image tokens, using a top-p sampling strategy with p set to 0.5. For decoding images using the diffusion model, the diffusion steps are set to 20. In evaluating the model, this paper compares the generated step images with real step images. Additionally, the paper evaluates by generating images using gold standard image tokens as an upper bound on model performance.

	SEED-LLaMA-8B	LaVIT-7B
Learning Rate	$1e-4$	$1e-5$
Optimizer	AdamW	
Weight Decay	0.05	0.1
Input Length	1024	
Batch Size	128	512

Table 3: Hyperparameter Settings

5.2 Baseline Methods

To compare with the proposed methods, we adopted a two-stage method from existing work as the baseline. In the first stage, we used the Vicuna-7B(Zheng et al., 2023) model to generate image captions. For a given step s_t, we concatenated the recipe title, description, ingredient list, and operation steps, and added the following instruction:

> Generate a detailed image caption for the step <s_t>.

This served as input, prompting the model to generate an image caption. In the second stage, we used the Stable Diffusion 2.1 model for text-to-image generation.

For the Vicuna-7B model in the first stage, we used image captions generated by the SEED-LLaMA model as the supervision signal and fine-tuned it with the same hyperparameters as the SEED-LLaMA. For the Stable Diffusion 2.1 model in the second stage, we fine-tuned it using the SEED-LLaMA model's image captions as text input. We trained it on the training set images with a learning rate of $1e-4$ and a batch size of 512 for 5,000 steps. During inference, we used the image captions generated by the Vicuna-7B model and input these captions into the Stable Diffusion 2.1 model for image generation, setting the diffusion steps to 20.

	Prec.	Rec.	F1
Single-step Image Generation (LaVIT)	62.0	52.3	56.7
+ *Explicit State Modeling*	71.2	62.2	66.4
Single-step Image Generation (SEED-LLaMA)	55.2	48.3	51.5
+ *Explicit State Modeling*	**76.2**	64.6	69.9
Step-by-step Image Generation (SEED-LLaMA)	64.8	56.4	60.3
+ *Explicit State Modeling*	75.1	**67.2**	**71.0**

Table 4: Human Evaluation Results of State Tracking.

Figure 3: Comparison of single-step image generation and step-by-step image generation.

5.3 Evaluation Metrics

To evaluate how well the generated step images model the entitiy states in the steps, we assessed the similarity between the model-generated step images and the real step images. Referring to the work of (Koh et al., 2023) and (Ge et al., 2024), we used similarity metrics based on the CLIP(Radford et al., 2021) model.

To evaluate the quality of the step images themselves, we employed the Fréchet Inception Distance (FID)(Heusel et al., 2018) metric.

6 Results

6.1 Quantitative Results

Baseline Method v.s. Ours Table 2 presents the CLIP similarity and FID metrics for the baseline and our proposed methods. Compared to the two-stage baseline, our methods significantly improve the CLIP similarity metric. Specifically, fine-tuning the SEED-LLaMA model with explicit state modeling in single-step image generation increases the CLIP similarity by over 10%. This indicates that multimodal generative models enhances the modelling of procedural text and images, thereby improving step image generation.

Single-Step v.s. Step-by-Step The step-by-step method outperforms the single-step method in both CLIP similarity and FID metrics. The step-by-step method's superior modeling and image generation quality likely result from incorporating information from previous images and sequential modeling, aligning better with the temporal nature of procedural texts.

Explicit State Modeling The FID metrics show that explicit state modeling significantly enhances image generation quality for both single-step and step methods. Generating an image caption before the image may make the task more similar to pre-training tasks, better utilizing the model's pre-trained text-to-image generation capabilities. For single-step generation, explicit state modeling improves CLIP similarity by 2.43%, possibly due to the "chain of thought" effect from generating a caption first.

Comparison of Base Models In single-step image generation and with explicit state modeling, LaVIT produces higher quality images (lower FID)

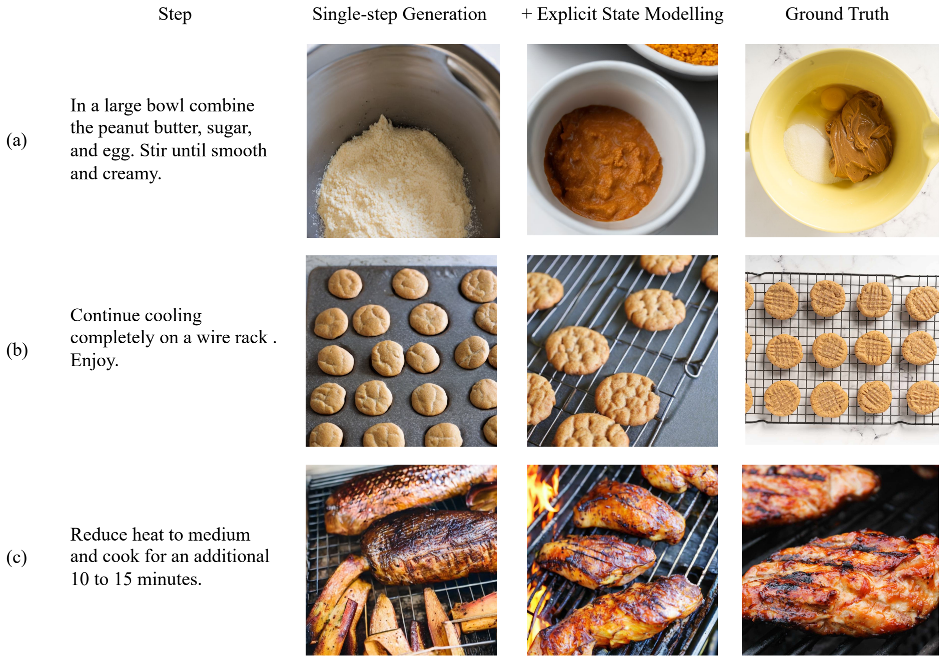

Figure 4: Comparison of single-step generation method and method with explicit state modeling.

and better tracks entity states (higher CLIP similarity) compared to SEED-LLaMA. LaVIT's use of longer token sequences captures more details, improving image quality. However, with explicit state modeling, SEED-LLaMA's CLIP similarity surpasses LaVIT's, likely due to SEED-LLaMA's instruction tuning, which generates more detailed image titles, better guiding image generation.

Comparison to Golden Image Tokens Despite improvements, current methods still show a significant gap in entity state modeling and image quality compared to images generated with gold standard image tokens, likely due to insufficient training data.

6.2 Entity State Tracking Analysis

To further analyze the effectiveness of entity state tracking of various methods, this paper sampled 20 cooking guides from the test set, consisting of a total of 138 step images, and manually evaluated the entity state tracking effectiveness of the step images generated by each method. For the generated images and the real images, the number of entities with correct states in the generated images and the total number of entities in the generated images and real images were manually counted to calculate Precision, Recall, and F1 scores.

Referring to existing works on the evaluation of state tracking in cooking guides (Amini et al., 2020; Zhang et al., 2021), this paper only considered whether the positional attributes were correct when evaluating whether the entity states were correct. When counting the number of entities, the same type of entity was counted only once. The results of the manual evaluation of state analysis are shown in Table 4, where Precision, Recall, and F1 are calculated using micro-averaging. The evaluation results show that explicit state modeling methods can significantly improve the entity state tracking effect, and the entity state tracking effect of the step-by-step image generation method is better than that of the single-step image generation method, which confirms the conclusions of the automatic evaluation of the entity state tracking effect of each method. In addition, the recall of each method is significantly lower than the precision, indicating that the model tends to generate a smaller number of entities when generating images.

6.3 Case Study

Single-step v.s. Step-by-step In the single-step image generation method, each step's image is generated independently, potentially causing inconsistencies where later images don't reflect information from earlier ones. The step-by-step image generation method addresses this issue. Figure 3 illustrates that in the single-step method, a baking rack present in an earlier step might be missing in a subsequent one. In contrast, the step-by-step method

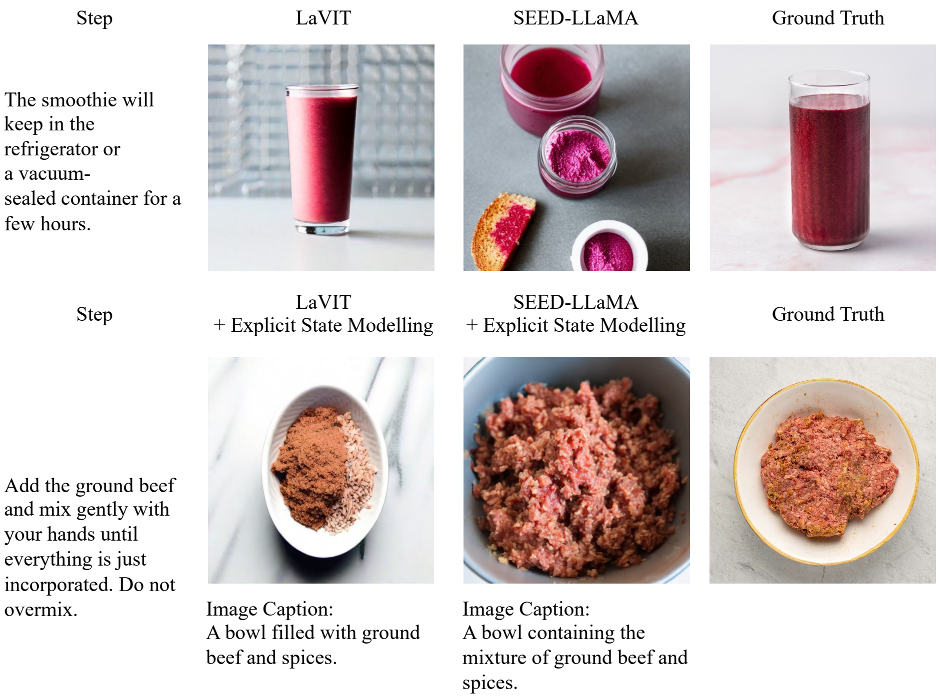

Figure 5: Comparison of base models.

allows the model to retain the baking rack information from the previous step, even if not explicitly mentioned in the current step's text.

Explicit Entity Modeling This paper manually evaluated images generated by a single-step method versus an improved method with explicit entity modeling. In 68% of 100 image pairs, the improved method yielded better results. The advantages include: 1) Correct Entity States: Explicit modeling improves accuracy in 17.6% of cases by accurately depicting entity states, such as accurately producing a mixture in a bowl in Figure 4(a); 2) No Missing Entities: It reduces omissions, correctly generating all entities like a wire rack in 23.5% of cases, such as including the wire rack in Figure 4(a); 3) High Image Quality: Leveraging pre-training capabilities, it enhances image quality in 58.8% of cases, such as producing undistorted chicken wings in Figure 4(c).

Comparison of Base Models This paper compared SEED-LLaMA and LaVIT base models in both single-step and improved methods. Figure 5 shows that in the single-step method, LaVIT better captures details like drink color, container shape, and background due to longer image token sequences. In the improved method, SEED-LLaMA generates more specific image titles, accurately depicting ingredient states, whereas LaVIT fails to do so, resulting in incorrect images. This indicates that SEED-LLaMA, after instruction fine-tuning, excels in generating accurate captions and step images.

7 Conclusion

In this paper, we introduced a novel task of step image generation in recipes, leveraging the step images as visual supervision for entity state tracking. By generating step images, we can visualize the entity states in each step. For this task, we collect a high-quality multimodal recipe dataset, theSpruceEats. Based on multimodal generative models, we proposed methods for both single-step and step-by-step image generation, incorporating explicit state modeling. Experiments on theSpruceEats dataset show that our methods enhance entity state tracking and image generation quality compared to existing methods.

Limitations

Our proposed method only involves the training of large language models and does not integrate the tokenization and diffusion modules of images into the joint training. This may result in sub-optimal quality of the generated images. If this issue can be addressed, there is potential for improving the quality of the step images generated by the model. Moreover, our proposed method is still far from truly reproducing the step images in real data. There are deficiencies in generating all entities and restoring entity states. In the future, there is significant room for improvement in terms of model design, training methods, and data scale.

References

Aida Amini, Antoine Bosselut, Bhavana Dalvi, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. *ArXiv*, abs/2003.13878.

Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *International Conference on Learning Representations*.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2019. Building dynamic knowledge graphs from text using machine reading comprehension. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. 2024. Planting a SEED of vision in large language model. In *The Twelfth International Conference on Learning Representations*.

Aditya Gupta and Greg Durrett. 2019. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Preprint*, arXiv:1706.08500.

Hao Huang, Xiubo Geng, Jian Pei, Guodong Long, and Daxin Jiang. 2021. Reasoning over entity-action-location graph for procedural text understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5100–5109, Online. Association for Computational Linguistics.

Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin CHEN, Chengru Song, dai meng, Di ZHANG, Wenwu Ou, Kun Gai, and Yadong MU. 2024. Unified language-vision pretraining in LLM with dynamic discrete visual tokenization. In *The Twelfth International Conference on Learning Representations*.

Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. 2023. Generating images with multimodal language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hugo Laurençon, Lucile Saulnier, Leo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. 2023. OBELICS: An open web-scale filtered dataset of interleaved image-text documents. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yunshui Li, Binyuan Hui, ZhiChao Yin, Min Yang, Fei Huang, and Yongbin Li. 2023. PaCE: Unified multi-modal dialogue pre-training with progressive and compositional experts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13402–13416, Toronto, Canada. Association for Computational Linguistics.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Hossein Rajaby Faghihi and Parisa Kordjamshidi. 2021. Time-stamped language model: Teaching language models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570, Online. Association for Computational Linguistics.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. *Preprint*, arXiv:2112.10752.

Jizhi Tang, Yansong Feng, and Dongyan Zhao. 2020. Understanding procedural text using interactive entity networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7281–7290, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium. Association for Computational Linguistics.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*, WWW '21. ACM.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023. Multimodal c4: An open, billion-scale corpus of images interleaved with text. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.