RecipeComics: Multi-Agent Orchestration for Sequential Instruction Visualization

Atharva Patil, Arnav Jhala

North Carolina State University, Department of Computer Science, Raleigh, North Carolina, USA

Abstract

Comic generation is an interesting challenge for multi-agent systems because it requires careful orchestration of layouts, economic use of text size, and composition of objects across image panels. As a representative example for multi-modal multi-agent orchestration, cooking recipes provides sequential multi-modal instruction domain with clear and precise evaluation metrics. These have been found to be useful in practical applications such as flight safety instructions, furniture assembly, and informational brochures. We present RecipeComics, an automated system transforming unstructured recipe text into multi-page comic books through multi-agent orchestration and custom image generation. The system addresses two key challenges: parsing variable text input through specialized LLM agents with structured schemas, and generating visually consistent sequential instruction panels. We introduce a quality framework defining six dimensions for instruction panel evaluation—four intra-image (structural realism, controlled hallucination, detail preservation, post-action state correctness) and two inter-image (environment consistency, style consistency)—with quantitative metrics using VQA and similarity-based approaches. Preliminary experiments on prompt structure variations demonstrate measurable impacts, with spatial and detail emphasis significantly outperforming background-first approaches. Our contributions include an end-to-end recipe-to-comic pipeline, a systematic quality framework for sequential instruction visualization, and experimental validation of design choices. The framework, including the experimental setup show promise for future work related to multi-modal agentic frameworks in other sequential generation tasks.

1. Introduction

Multi-agent orchestration and communication across agents that combine varied tasks with different representations/formats, and modalities like visual, auditory, and textual is a current area of research focus in the community, particularly for practical applications. To explore the challenges of combining a variety of models for planning, search, visual representation, composition, and balance between text and visual modalities, we explore the area of multi-agent comic generation [1].

Text-based cooking recipes create challenges with processing speed, cognitive load during active cooking, and accessibility for diverse learners. Comics offer an alternative format that addresses these challenges through sequential visual storytelling while enabling faster step lookup and reducing language barriers. However, manual creation requires both artistic skill and culinary knowledge, making it impractical at scale. We present RecipeComics[1], an automated system transforming unstructured recipe text into multi-page comic books with three components: a cover page representative of the final dish, ingredient pages displaying cards in grid layouts, and instruction pages with sequential panels illustrating cooking steps1.

This transformation poses two key challenges. First, recipe text varies dramatically—from well-formatted lists to informal descriptions with ambiguous phrasing leading to ill-defined goals for agents. The team of agents must parse variable input, extract structured information, and make editorial decisions for downstream consistency leading to orchestration of agent communication protocols, schemas, and agreement on consistency. Second, generating visually consistent sequential instruction images requires both high-quality individual images and cross-sequence congruity. Unlike standalone generation, sequential visualization demands narrative continuity both in the visual and textual domains. This relates to the challenge of planning layouts based on goals specified in structured discourse [2].

Joint Proceedings of the ACM IUI Workshops 2026, March 23-26, 2026, Paphos, Cyprus

aspatil2@ncsu.edu (A. Patil); ahjhala@ncsu.edu (A. Jhala)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]The complete CrewAI based implementation is publicly available at: https://github.com/atharvapatil22/crewai_recipe_comic_generator

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-06-27

CEUR-WS.org/Vol-4266/paper05.pdf

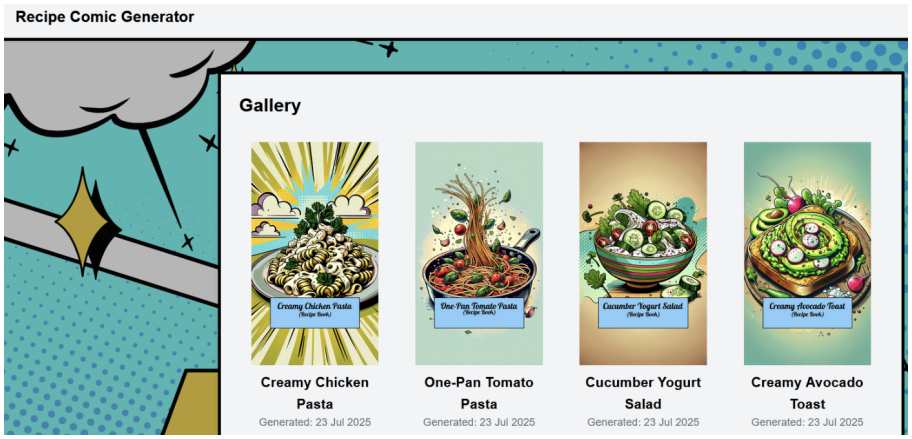

Figure 1: Some recipe comics generated by the RecipeComics system.

We address these challenges through the development of a multi-agent framework (with CrewAI as the software library) for specialized LLM language agents and ComfyUI sequential image workflows with Stable Diffusion for image generation. The human intervention is in curating narrative schemas for story generation, and prompt and parameter tuning in the ComfyUI workflows. Agents operate with structured output schemas to manage stochasticity across focused sub-tasks: validation, extraction, prompt generation, and image generation. We contribute (1) an end-to-end recipe-to-comic pipeline demonstrating multi-agent orchestration, (2) a quality optimization framework defining six dimensions—four intra-image (structural realism, controlled hallucination, detail preservation, post-action state correctness) and two inter-image (environment consistency, style consistency)—with quantitative evaluation metrics, and (3) preliminary experiments showing spatial and detail emphasis prompts outperform background-first approaches. The framework is designed to be generalizable to other sequential procedural instruction domain requiring visual consistency across steps through definition of roles for each module and communication schemas between agents.

2. Related Work

Recent advances in large language models have enabled multi-agent systems where specialized agents collaborate on complex tasks. Li et al. propose TDAG, a framework decomposing tasks into subtasks assigned to specialized subagents, reducing noise from irrelevant information [3]. Guo et al. survey LLM-based multi-agent systems, identifying orchestration as critical for scalability and emphasizing optimized workflows [4]. Our work applies these principles to recipe-to-comic transformation, employing specialized agents with structured output schemas for validation, extraction, and generation.

Generating visually consistent sequential images for storytelling presents unique challenges. Chen and Jhala integrate narrative theories with generative models, using semantic mapping to ensure consistency across comic panel sequences [5]. Shin et al. centralize story elements in narrative graphs with canonical visual attributes, enabling LLMs to generate prompts maintaining consistent appearance [6]. While these works emphasize character consistency, their approaches to environmental and object consistency inform our framework for procedural instruction visualization, where kitchen environments and artistic style must remain uniform across sequences.

Evaluating text-to-image generation requires metrics beyond single-image assessment. Cho et al. propose VPGen and VPEval frameworks, decomposing generation into object/count, layout, and image steps, with evaluation programs providing interpretable explanations [7]. Our evaluation framework adapts these modular approaches, defining six dimensions—four intra-image and two inter-image—with quantitative metrics for both individual panel quality and sequence-level consistency. Our work distinguishes itself by combining multi-agent orchestration with a systematic quality optimization framework specifically for procedural instruction visualization.

3. Pipeline Overview

The RecipeComics pipeline transforms dynamic textual input into a comic book output through three stages: preprocessing, panel generation, and composition. The system combines specialized LLM-based agents, image generation models, and deterministic processing modules. Two design principles guide our architecture: (1) *agent granularity*—decomposing complex tasks into focused sub-tasks improves LLM performance by reducing cognitive load and minimizing irrelevant information [3], and (2) *structured schemas*—enforcing Pydantic output schemas ensures consistency and reliability by minimizing errors in LLM outputs [8]. This combination manages LLM stochasticity while ensuring reliable data flow between pipeline stages. Figure 2 shows the complete pipeline overview.

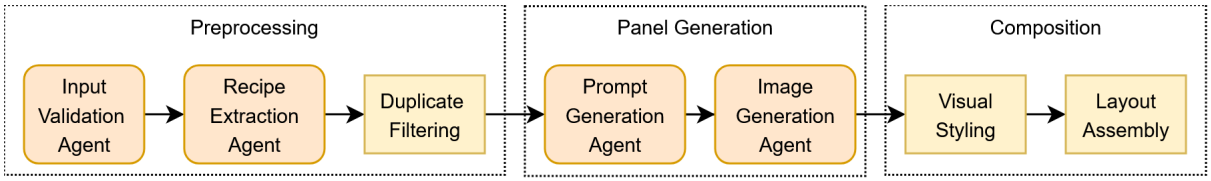

Figure 2: RecipeComics pipeline overview

3.1. Preprocessing Stage

Input Validation Agent — This agent determines whether the input text resembles a valid recipe. Validation criteria require a logical instruction sequence and clear ingredient mentions (either as a separate list or embedded in instructions). If validation fails, the pipeline terminates and provides user feedback. This validation ensures that only recipe-like inputs proceed to downstream agents, preventing the production of meaningless artifacts from irrelevant text.

Recipe Extraction Agent — This agent transforms raw text into a structured format conforming to a Pydantic schema with recipe name, ingredient list (name and quantity pairs), and instruction list. User inputs can vary from well-formatted recipes to informal conversational text. The agent makes editorial decisions if required: inferring missing recipe titles, cleaning ingredient quantities to remove preparation details, resolving alternative ingredients to definitive selections (to eliminate ambiguity in downstream image generation), and inferring missing quantities based on typical serving sizes. The output serves as the canonical representation for all subsequent stages.

Duplicate Filtering — A deterministic processing module compares the current recipe against previously generated comics using weighted similarity scoring (70% recipe name string similarity, 30% ingredient Jaccard overlap). When similarity exceeds 0.80, users can opt to reuse an existing comic rather than generating new ones. This optimization is particularly valuable for frequently submitted recipes that exhibit minor textual variations despite identical core content.

3.2. Panel Generation Phase

Prompt Generation Agent — This agent creates image generation prompts from structured recipe data. It generates a prompt for the cover image, one prompt per ingredient, and one prompt per instruction step. Prompts for ingredients describe minimalist presentations suitable for standalone cards. Prompts for instruction panels describe kitchen scenes corresponding to the textual instruction with relevant cookware and environmental context. All prompts include comic book style specifications.

Image Generation Agent — This agent executes prompts through generative models to produce panels for all three page types. Cover pages use DALL-E 3 for representative imagery, while ingredient and instruction images use JuggernautXL with LoRA models for style control through ComfyUI workflows (parameters in Appendix B). The quality optimization framework for instruction panels, including evaluation dimensions and systematic variations, is detailed in Section 4.

3.3. Composition Stage

Visual Styling — A deterministic processing module applies post-processing to generated images using Python PIL. The ingredient images are placed on styled cards with name and quantity labels. The instruction panels are framed with the corresponding textual instruction. Cover images receive title card overlays. This styling enhances the readability of the final comic and establishes a consistent visual identity across system outputs.

Layout Assembly — Final comics follow a fixed layout: a full-page cover with title overlay, ingredient pages with 12-card grids (scaling with ingredient count), and instruction pages with 3 vertically stacked panels per page (scaling with instruction count). This layout prioritizes clarity and readability; dynamic layouts are discussed in Section 6.

3.4. Technical Implementation

The RecipeComics implementation is an event-driven distributed system. A Flask orchestrator service exposes REST APIs and manages Redis message queues (preprocess, core-generation). A preprocessing service handles validation, extraction, and duplicate filtering; a core-generation service handles prompt generation, image generation, visual styling, and layout assembly. A Next.js frontend provides the user interface. LLM agents use GPT-4o-mini orchestrated via CrewAI. Cover images use DALL-E 3, while ingredient and instruction images use JuggernautXL through ComfyUI workflows with specific parameters detailed in Appendix B. Complete implementation details are available in the public repository (Appendix A). This architecture enables independent scaling and asynchronous processing.

4. Framework for Instruction Panels

Panel generation involves stochastic models—LLMs for prompt creation and diffusion models for image synthesis—exhibiting unpredictable behavior. Just as we enforce Pydantic schemas on agent outputs to ensure structural consistency, we define quality criteria for generated images to standardize outputs despite model variability. Instruction panels specifically require narrative congruity as they are sequential procedural visuals advancing cooking workflows within continuous kitchen environments. Readers expect consistent kitchen spaces with uniform lighting, décor, and spatial layout across panels. While designed to provide structural guidance for instruction panels in our recipe-to-comic pipeline, this framework can be generalized to any domain requiring sequential image generation with environmental consistency.

4.1. Quality and Consistency Dimensions

This framework defines six dimensions representing ideal attributes that generated images should exhibit. We categorize these as intra-image dimensions (applicable to individual panels) and inter-image dimensions (applicable to sequences).

Intra-Image Quality Dimensions:

Structural Realism: Physically plausible structures and spatial relationships without object merging, physics violations, or structural anomalies.

Controlled Hallucination: Visual attributes are applied only to intended objects, without semantic bleeding to unintended elements. And object counts remain reasonable without excessive background duplication.

Detail Preservation: Visual attributes explicitly specified in recipe instructions accurately rendered, including shape modifications, color characteristics, and textural properties

Post-Action State Correctness: Completed action states are depicted with visible outcomes rather than intermediate or in-progress states. Since recipe instructions describe temporal processes, panels should show the post-action state where the outcomes are visually evident.

Inter-Image Consistency Dimensions:

Kitchen Environment Consistency: Uniform background elements throughout sequences, including appliances, cookware placement, furniture, décor, and color schemes.

Artistic Style Consistency: Uniform visual style throughout sequences, including shading, lighting direction, line work, and comic aesthetic characteristics.

4.2. Modules of the Panel Generation Stage

Panel generation is an intermediate stage within the broader pipeline with defined input and output specifications. For instruction panels, the input is a list of recipe instruction sentences extracted by the Recipe Extraction Agent (Section 3). The output is a corresponding list of images—one per instruction—satisfying the quality and consistency dimensions above. Figure 3 illustrates the refined structure of the panel generation stage.

Prompt Generation Module converts instruction sentences into text prompts for image generation. This module receives structured recipe data and produces detailed prompts describing kitchen scenes, relevant cookware, environmental context, and comic book style specifications.

Baseline Image Generation Module executes text-to-image generation using diffusion models. This module processes prompts through the generative model to produce instruction panel images. Our implementation uses ComfyUI workflows with Stable Diffusion models and LoRA adapters for style control.

Consistency Enhancement Module applies optional post-processing or conditioning techniques to improve inter-image consistency. While baseline generation produces images independently (each conditioned only on its prompt), this module can incorporate additional constraints or information to improve sequential congruity.

Systematic exploration of variations across these modules can identify configurations that optimize alignment with the quality and consistency dimensions defined in Section 4.1.

4.3. Proposed Variations

Alternative approaches within each module represent an exploration space for optimizing panel quality. Prompt generation variations can emphasize different compositional aspects—spatial context, color/texture detail, semantic ordering, or punctuation density—directly affecting how generation models interpret instructions. Table 1 presents eight structural variations for systematic comparison. Baseline image generation quality depends on model architecture and parameter configuration. Different diffusion models exhibit varying strengths in scene composition and detail rendering, while CFG scale and sampling steps trade prompt adherence against visual naturalness, impacting structural realism and detail preservation. Consistency enhancement approaches—such as conditioning on previous panels via IP adapters or compositing foreground elements onto standardized backgrounds—can improve environmental and style uniformity across sequences. Our preliminary experiments focus on prompt variations using JuggernautXL (parameters in Appendix B), with baseline and consistency variations being directions for future investigation.

5. Preliminary Experiments and Results

As initial validation of our framework, we present preliminary experiments focusing on prompt generation variations—the first of three identified variation points in Section 4.2.

5.1. Dataset

We have designed a dataset of 18 recipes spanning six dish categories, each selected for distinct visual and technical characteristics including assembly-focused preparations, cooking-intensive processes, liquid-based transformations, and baking workflows. Recipes were generated via LLMs and manually cleaned and verified for clarity, completeness, and alignment with typical culinary workflows. As our work progresses, this dataset with diverse dish types will enable comprehensive evaluation of the

Table 1
Prompt Generation Variations

Recipe instruction (common to all examples): "Slice the tomatoes into rounds"

Type	Label	Description	Generated Prompt Example
Minimal	a	Core objects and action results only, minimal descriptors	tomato rounds on cutting board, knife beside, simple kitchen background, natural lighting, comic book illustration, vintage comic art style, bold ink outlines, cel shaded
Spatial & Bg.	b	Spatial placement and context, minimal color/texture	cutting board placed on countertop, tomato rounds arranged on board, knife resting to the right, small bowl nearby, simple kitchen background, natural lighting, comic book illustration
Color & Texture	c	Detailed color/texture, minimal spatial context	bright red tomato rounds on cutting board, glossy cut surfaces, juicy flesh visible, stainless steel knife, simple kitchen background, natural lighting, comic book illustration
Max Detail	d	Combines spatial, color/texture, and context	cutting board on countertop, bright red tomato rounds arranged neatly, shiny stainless steel knife to the right, small bowl behind board, simple kitchen background, natural lighting
Ingredient-First	e	Ingredients/food before cookware/tools	tomato rounds on a cutting board, knife on side, simple kitchen background, natural lighting, comic book illustration, vintage comic art style, bold ink outlines
Cookware-First	f	Cookware/tools before ingredients	cutting board with tomato rounds, knife beside, simple kitchen background, natural lighting, comic book illustration, vintage comic art style, bold ink outlines
Background-First	g	Scene description precedes main objects	simple kitchen background, natural lighting, comic book illustration, vintage comic art style, bold ink outlines, cel shaded, cutting board on counter with tomato rounds, knife beside
Sparse Punct.	h	Reduced commas, longer phrases	cutting board with tomato rounds and knife beside on counter, simple kitchen background with natural lighting, comic book illustration in vintage comic art style

Note: All prompts include standard style specifications (comic book illustration, vintage art style, etc.)

Instruction Panel Framework across varied cooking methods, appliance usage, and visual complexity in future work.

For the preliminary experiments reported here, we focus on three salad recipes from this dataset. Each recipe contains 8-9 instruction steps. Salads provide a suitable initial evaluation target as they involve multiple preparation techniques (chopping, mixing, assembly) with visually distinct ingredients and clear action results.

5.2. Evaluation Metrics

For preliminary evaluation, we employ a set of metrics to assess the intra-image quality dimensions defined in Section 4.1. All metrics are computed using pretrained models: BLIP-VQA-base for visual question answering [9], CLIP-ViT-Base-Patch32 for image-text similarity [10], and SAM for image segmentation [11]. DINO-ViT embeddings are used for style similarity measurements [12].

List of metrics:

CLIP Score measures alignment between generated images and their generation prompts. We compute cosine similarity between CLIP image embeddings and text embeddings of the corresponding prompt, normalized to [0,1].

Action Result Score measures whether panels depict completed action states. VQA question: "Does

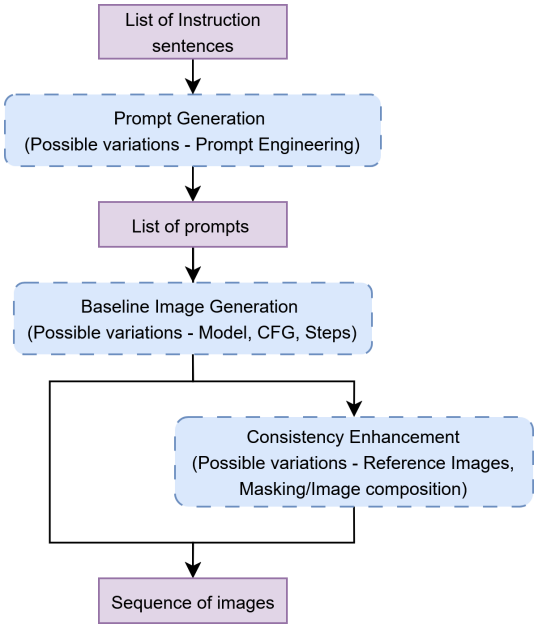

Figure 3: Panel Generation Stage

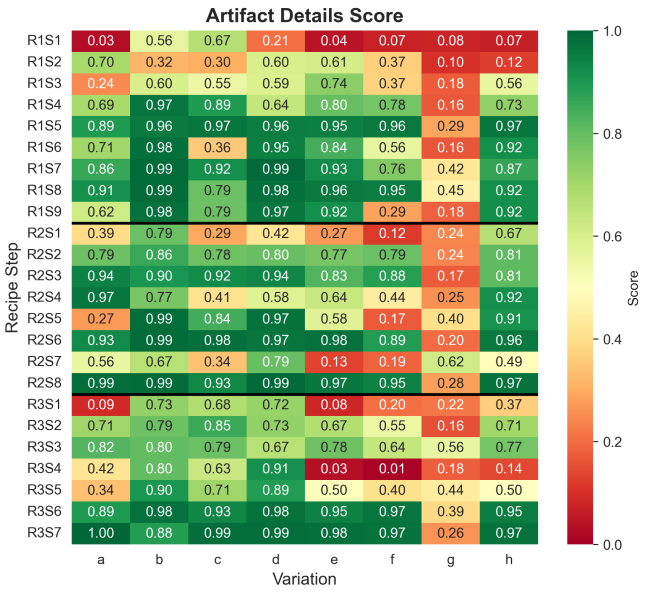

Figure 4: Detail Preservation Score heatmap

this image show the completed result of '[recipe_step]'?" We extract the probability of a "yes" response from the VQA model's output distribution.

Detail Preservation Score measures whether specified details appear. VQA question: "Does this image contain the specific details of shape, color and texture mentioned in:'[recipe_step]'?"

Controlled Hallucination Score measures appropriate object presence. VQA question: "Are there excessive duplicates or too many repeated objects bunched together in the background?" Scoring is inverted (1 - raw probability) so higher values indicate better control.

For VQA-based metrics, we use BLIP to generate yes/no responses and extract probabilities by computing softmax over the output vocabulary, then normalizing the yes/no token probabilities. All metrics yield scores in [0,1]. We validated metric sensitivity by benchmarking: white images produce scores near 0, while gold-standard kitchen images produce scores near 1.

These metrics provide initial assessment of the quality dimensions for our preliminary experiments. Comprehensive evaluation would incorporate additional complementary metrics—such as object detection for counting verification, segmentation-based structural analysis, and fine-grained attribute classifiers—to more thoroughly capture all aspects of the defined dimensions. Such extensions are discussed in Section 6.

5.3. Evaluation and Results Discussion

Setup. We evaluated the eight prompt variations defined in Section 4.3 on three salad recipes. Recipe 1 (R1) contains 9 instruction steps, Recipe 2 (R2) contains 8 steps, and Recipe 3 (R3) contains 7 steps, yielding 24 instruction steps total. For each step, we generated images using all eight prompt variations (a-h corresponding to the types in Table 1), producing 192 images total. Notation R(n)S(m) denotes Recipe n, Step m (e.g., R2S5 represents the fifth step of the second recipe). Figure 5 presents aggregated metric scores across all variations.

CLIP Score. CLIP scores remain remarkably stable across all variations, clustering tightly in the 0.64-0.69 range with minimal variation between prompt types. This stability suggests that all prompt structures successfully communicate core semantic content to the generation model—the fundamental subject matter (ingredients, actions, kitchen setting) is consistently represented regardless of prompt variations. The limited sensitivity indicates that CLIP embeddings capture coarse semantic alignment but may not reflect fine-grained differences in detail preservation or compositional quality that distinguish the prompt variations.

Action Result Score. Action Result scores exhibit substantial variability both across prompt variations and across recipe steps. Figure 4 reveals step-specific patterns: certain steps (e.g., R2S5, R2S6, R3S4) consistently achieve high scores (0.8+) across nearly all variations, while others (e.g., R1S1, R2S7, R3S3) score poorly (<0.3) regardless of prompt structure. This step-dependency suggests that depicting completed action states depends heavily on the inherent visual clarity of the recipe instruction itself—steps describing visually distinctive outcomes (e.g., "mix until well combined" showing uniform texture) are easier to represent than ambiguous states. Variations b (Spatial & Background) and d (Max Detail) show slightly higher average scores, likely because explicit spatial and contextual information helps the model frame the post-action state more accurately.

Detail Preservation Score. Detail Preservation shows the highest sensitivity to prompt structure, as evidenced by substantial score differences across variations in Figure 4. Variation g (Background-First) consistently underperforms, often scoring below 0.3 even on steps where other variations exceed 0.9. This suggests that prioritizing scene description before main objects causes the model to allocate attention to environmental elements at the expense of ingredient-specific details (shape, color, texture). Conversely, variations b, d, and h maintain strong detail preservation (often 0.8+), indicating that explicit color/texture descriptors or comprehensive detail inclusion successfully guides the model to render specified visual attributes. The variability across steps reflects the difficulty of preserving fine-grained details—steps involving distinctive visual characteristics (e.g., "golden brown onions", "finely diced tomatoes") show greater prompt sensitivity.

Controlled Hallucination Score. Scores for controlled hallucination remain generally high (0.7-0.95) across most variations and steps, indicating that the generation model successfully avoids excessive object duplication and inappropriate color bleeding in most cases. This relatively consistent performance suggests that the base model (JuggernautXL) and generation parameters (7.5 CFG, 22 steps) provide adequate control over hallucination tendencies, with prompt structure playing a secondary role. Variation c (Color & Texture) shows slightly more variability, potentially because detailed color specifications occasionally cause unintended propagation of color attributes to background elements.

Overall Findings. No single prompt variation dominates across all metrics, suggesting inherent trade-offs in prompt design (see Figure 5).

Variation g (Background-First) consistently underperforms, particularly on detail preservation, indicating that leading with scene description dilutes attention to critical foreground elements. Variations b (Spatial & Background), d (Max Detail), and h (Sparse Punctuation) demonstrate balanced performance across metrics, with d showing slight advantages on action result and detail scores, likely due to its comprehensive inclusion of spatial, color, and contextual information. The minimal variation (a) performs adequately but shows lower detail preservation, confirming that explicit descriptors improve fine-grained visual fidelity. These findings validate the framework's utility in identifying effective prompt strategies: instruction panel generation benefits from prompts that balance spatial context with explicit detail specifications while maintaining focus on primary objects rather than scene-setting elements.

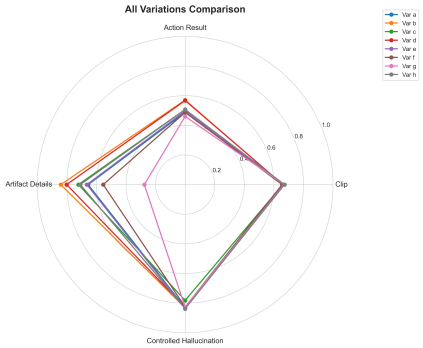

Figure 5: Comparison of all prompt variations across four intra-image quality metrics

6. Ongoing and Future Work

6.1. System Extensions

The current composition stage employs fixed layouts (three vertically stacked rectangular panels per page), limiting expressive potential. A layout-first approach introducing a layout generation agent before panel generation could analyze instruction sequences to design adaptive layouts based on step importance and complexity—allocating full-width panels to crucial steps while minor actions occupy compact side panels. This would enable varied compositional structures (diagonal divisions, L-shaped

panels, variable panel counts) accommodating instruction density while creating visual hierarchy. The restructured workflow would shift panel generation to filling predetermined regions defined by layout templates with spatial masks, potentially improving composition through boundary constraints and enhancing pedagogical effectiveness through dynamic visual emphasis.

Additionally, current pipeline agents performing multiple subtasks could benefit from further decomposition. The Recipe Extraction Agent, for instance, simultaneously handles structural transformation, editorial modifications, and content filtering. Decomposing into specialized agents would enable independent optimization per subtask, following the agent granularity principle (Section 3) where narrower scope reduces LLM cognitive load and allows targeted prompt engineering.

Furthermore, deployment optimizations could substantially reduce cost and latency. Replacing commercial LLM APIs (GPT-4o-mini) with locally hosted open-source models (Llama 3, Mistral) would eliminate per-token costs and API overhead, improving efficiency and response times. The distributed architecture could be refined with dedicated worker services for computationally intensive stages, enabling stage-specific scaling—panel generation could scale to multiple parallel workers while lightweight stages operate with minimal resources.

6.2. Evaluation

Comprehensive assessment of the complete RecipeComic system requires evaluation beyond individual component performance. User studies will be essential for measuring practical utility: recipe comprehension rates, perceived visual quality, narrative consistency, and overall usefulness compared to text-only formats. Task-based studies could measure cooking success rates and completion times using generated comics versus traditional recipes. System efficiency metrics—end-to-end processing time, computational resource requirements, and cost analysis across model choices—would inform deployment decisions. Ablation studies isolating individual components would quantify their contributions to output quality and efficiency.

Additionally, the preliminary experiments in Section 5 represent initial validation of one variation point within the framework for instruction panels. Comprehensive evaluation could extend to systematic exploration across all three modules across diverse dish categories. Testing across the full six-category dataset (salads, tacos, pastas, curries/stews, smoothies, baking) would validate framework robustness across varied visual characteristics—raw versus cooked ingredients, liquid transformations, different appliance usage, and diverse temporal processes.

7. Conclusion

This paper presents RecipeComics, an end-to-end system for transforming unstructured recipe text into multi-page comic books through multi-agent orchestration and custom image generation. Our primary contributions are threefold: (1) a complete recipe-to-comic pipeline demonstrating practical application of specialized LLM agents and image generation models combined with deterministic processing modules, (2) a systematic quality framework defining six dimensions for evaluating instruction panel generation with focus on both individual image quality and sequential consistency, and (3) preliminary experimental results showing measurable impacts of prompt structure methods on panel quality. By bridging practical system engineering with systematic quality control for generative AI outputs, this work demonstrates an approach to building reliable multi-modal pipelines where stochastic models must produce consistent, high-quality results.

The multi-agent orchestration approach demonstrates a pattern of content transformation tasks where unstructured input must be parsed, validated, and transformed into structured multi-modal outputs based on specific communicative goals and composition constraints. The quality framework for instruction panels potentially generalizes to other sequential instruction visualization domains through an example of defining both general and domain-specific evaluation metrics. Assembly instructions for furniture or electronics, educational science experiment tutorials, DIY home repair guides, and craft project instructions all share some of the requirements for visually consistent sequential panels depicting procedural steps. The quality and consistency dimensions defined in Section 4.1 apply directly to these

domains with minimal adaptation. The systematic evaluation methodology provides a transferable approach to assessing visual quality in procedural image generation tasks.

Declaration on Generative AI

During the preparation of this work, the authors used Stable Diffusion base model Juggernaul-XL v9 and Dall-E 3 to Generate images. GPT-4o-mini model was used for part of the agentic workflows mentioned in the paper. This model was also used for some Grammar and spelling checks, and for paraphrasing and rewording sentences. The authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Y.-C. Chen, A. Jhala, Narrative-to-scene generation: An llm-driven pipeline for 2d game environments, in: Joint AIIDE Workshop on Experimental Artificial Intelligence in Games and Intelligent Narrative Technologies, November 10-11, 2025, Edmonton, Canada., 2025.

[2] A. Jhala, R. M. Young, Cinematic visual discourse: Representation, generation, and evaluation, IEEE Transactions on computational intelligence and AI in games 2 (2010) 69–81.

[3] Y. Li, J. Liu, Y. Sun, H. Su, N. Al Moubayed, T. Breckon, Tdag: A multi-agent framework based on dynamic task decomposition and agent generation, Neural Networks 185 (2025) 106937.

[4] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. Chawla, O. Wiest, X. Zhang, Large language model based multi-agents: A survey of progress and challenges, in: Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI), 2024.

[5] C.-Y. Chen, A. Jhala, Collaborative comic generation: Integrating visual narrative theories with ai models for enhanced creativity, in: CREAI 2024: International Workshop on Artificial Intelligence and Creativity, ECAI, 2024.

[6] S. Shin, J. Kim, H. Lee, S. Park, Generating visually consistent images for storytelling via narrative graph prompting, in: ICCV 2025 Workshop on AI for Creative Storytelling (AISTORY), 2025.

[7] J. Cho, A. Zala, M. Bansal, Visual programming for text-to-image generation and evaluation, in: Advances in Neural Information Processing Systems (NeurIPS), volume 36, 2023.

[8] H. Chase, Langchain: Building applications with llms through composability, GitHub repository, 2022. Available at: https://github.com/langchain-ai/langchain.

[9] J. Li, D. Li, C. Xiong, S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: ICML, 2022.

[10] A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: ICML, 2021.

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. Berg, W. Lo, P. Dollár, R. Girshick, Segment anything, in: ICCV, 2023.

[12] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: ICCV, 2021.

A. Image Generation Hyperparameters

Parameter	Value
Base Model	Juggernaut-XL v9
LoRA	EldritchComicsXL 1.2 (strength: 0.9)
Sampler	DPM++ 2M SDE Karras
Steps	22
CFG Scale	7.5
Resolution	1344 × 768 pixels

Table 2
ComfyUI workflow parameters for instruction panel generation