MangaFlow: An End-to-End Agentic Framework for Controllable Story to Manga Generation

Muyao Wang
The University of Tokyo
muyaowang@g.ecc.u-tokyo.ac.jp

Zeke Xie*
Hong Kong University of Science and Technology (Guangzhou)
zekexie@hkust-gz.edu.cn

Yanhao Chen
The University of Tokyo
chen-yanhao@g.ecc.u-tokyo.ac.jp

Lixin Xiu
The University of Tokyo
xiu-lixin106@g.ecc.u-tokyo.ac.jp

Hideki Nakayama
The University of Tokyo
nakayama@ci.i.u-tokyo.ac.jp

Abstract

End-to-end manga generation is a structured visual storytelling task that requires story decomposition, recurring character and scene grounding, page layout design, panel rendering, page composition, and lettering. However, existing generative models often perform direct page synthesis, entangling these factors in a single visual output and limiting precise control over layout geometry, visual references, and cross-panel consistency. To address these limitations, we propose **MangaFlow**, an agentic framework for controllable long-form manga generation that decomposes manga creation into planning, grounding, layout construction, reference-conditioned rendering, composition, and text placement. By treating layout and visual references as explicit intermediate variables, MangaFlow enables both simple text-to-manga generation and more precise user-controlled manga creation. This design exposes layout, visual assets, and lettering as editable intermediate controls for refining panel geometry, references, and text placement. To support long-form consistency, MangaFlow introduces a *story section memory* that links section descriptions with corresponding character, scene, and object references for reuse across panels. We further present a meta-benchmark for evaluating layout controllability, visual consistency, and generation quality. Experiments show that MangaFlow improves layout adherence and cross-panel consistency over direct generation baselines while supporting flexible human control. Our code is available at MangaFlow.

1 Introduction

Manga generation is an emerging form of language-guided visual storytelling that goes beyond single-image synthesis. It requires a complete production process involving narrative decomposition, character and scene consistency, page-level panel layout, panel-wise rendering, and dialogue or narration placement (Wu et al., 2025; Vivoli et al., 2024b; Chen et al., 2024). Recent story visualization methods have explored coherent image sequence generation from textual narratives (Maharana et al., 2022; Chen et al., 2022; Zhou et al., 2024; Rahman et al., 2023; Li et al., 2019), and manga generation methods have begun to address multi-panel page structures (Chen et al., 2024). However, existing approaches still provide limited support for controllable end-to-end manga production, especially for editable layouts and lettering, reusable character and scene references, and structured consistency control across panels and pages.

A central difficulty is that manga layout is not merely a visual attribute, but a core narrative device (Feng et al., 2024; Cao et al., 2012; Jing et al., 2015). Panel size, arrangement, reading order, and empty space directly affect pacing and visual emphasis. Yet direct page generation methods usually treat layout as an implicit outcome of image synthesis (Xie et al., 2023; Qu et al., 2023; Inoue et al., 2023). As a result, they may fail to follow specified panel geometry, produce incorrect panel counts, merge panel boundaries, or ignore creator-defined layout constraints. Since artists typically design page structure before drawing panels, manga generation systems should treat layout as an explicit, editable, and enforceable structural variable.

Large Language Model (LLM)-based agents offer a natural way to decompose such a complex generation process. Prior agentic systems have shown that language models can support planning, reasoning, tool use, and multi-step coordination (Yao et al., 2022; Wu et al., 2024), while visual generation agents improve controllability by decomposing complex instructions into specialized subtasks (Wang et al., 2024; Jin and Song, 2023; Zhou et al., 2025; Kim et al., 2025). Manga generation similarly requires transforming a single story prompt into structured decisions over story planning, scene segmentation, character grounding,

*Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 5985–6002
October 24-29, 2026 ©2026 Association for Computational Linguistics



Figure 1: Example of end-to-end text-to-manga generation by MangaFlow. The generated manga consists of four pages, and each page should be read from right to left following the standard manga reading order.

layout construction, panel rendering, page composition, and lettering. However, generic agent pipelines are insufficient: without task-specific memory and layout control, information may decay across stages, causing character drift, scene inconsistency, object inconsistency, or layout violations.

To address these challenges, we propose **MangaFlow**, an end-to-end agentic framework for controllable manga generation. To the best of our knowledge, MangaFlow is the first agentic system designed for controllable end-to-end manga generation. MangaFlow decomposes manga creation into narrative planning, story-section construction, explicit layout generation, reference-conditioned panel rendering, page composition, and lettering. MangaFlow has two key designs: story section memory and explicit layout control. The former links coherent story sections with fixed textual descriptions and visual references for recurring characters, scenes, and key objects, enabling consistent reuse across panels. The latter treats layout as a first-class control variable, supporting manual layout specification, reference-layout reuse, and editable intermediate assets.

In addition, evaluation is another open challenge. Existing visual storytelling benchmarks mainly focus on character consistency, prompt alignment, or visual quality (Zhuang et al., 2025; Gao et al., 2025), while end-to-end manga generation also requires evaluating layout controllability, panel count correctness, lettering quality, and adherence to user-provided constraints. We therefore introduce **MangaGen-MetaBench**, a meta-benchmark covering layout-controlled generation, and story-section-guided generation. Rather than serving as a final comprehensive benchmark, it provides a practical protocol for exposing the limitations of direct generation methods and validating the controllability of MangaFlow.

Our contributions are summarized as follows:

- We formulate controllable end-to-end manga generation as a structured visual storytelling task requiring narrative planning, layout control, visual consistency, and lettering.
- We propose **MangaFlow**, an agentic framework that supports explicit layout control, reference-conditioned panel rendering, and editable intermediate assets, with story section memory binding section descriptions to character, scene, and object references for cross-panel consistency.
- We propose **MangaGen-MetaBench**, a meta-benchmark for evaluating manga-specific layout controllability, consistency, generation quality, and lettering.

2 Related Work

Manga Generation Early manga and comic generation studies mainly focus on layout design, page composition, or transforming existing visual materials into comic-style pages. For example, automatic manga layout and video-to-comic systems explicitly model panels, page composition, and text balloons, showing that manga creation requires more than independent image synthesis (Cao et al., 2012; Jing et al., 2015; Yang et al., 2021). Large-scale manga resources and understanding benchmarks, such as Manga109 (Aizawa et al., 2020), Manga109Dialog (Li et al., 2024), and CoMix (Vivoli et al., 2024a), further provide annotations for panels, characters, text, reading order, speaker association, and page-level understanding.

More recently, MangaDiffusion (Chen et al., 2024) formulates text-only manga generation and constructs Manga109Story to study multi-panel manga page generation from plain text, while DiffSensei (Wu et al., 2025) introduces customized manga generation with multi-character control and layout-aware character adaptation. These works mostly focus on isolated components such as layout conditioning, character customization, or page generation. MangaFlow targets the missing setting of end-to-end controllable manga creation with an editable intermediate pipeline.

Story Visualization Story visualization aims to generate a sequence of images from a textual narrative and has been studied as a natural extension of text-to-image generation. Early methods such as StoryGAN (Li et al., 2019) and StoryDALL-E (Maharana et al., 2022) establish the basic formulation of mapping multi-sentence stories to coherent image sequences, while later methods improve visual consistency through memory, planning, attention, or diffusion-based generation (Mao et al.; Meng et al.; Zhang et al., 2025; Rahman et al., 2023; Zhou et al., 2024; Chen et al., 2022; Shen and Elhoseiny, 2025; He et al., 2025). Recent benchmarks such as VinaBench (Gao et al., 2025) and ViStoryBench (Zhuang et al., 2025) further evaluate story visualization from perspectives such as narrative coherence, character consistency, visual style, and prompt alignment. MangaFlow adapts the long-range consistency goal to manga by organizing panels into story sections.

Agent Systems LLM-based agent systems have shown strong potential for decomposing complex tasks into planning, reasoning, tool use, and multi-step execution (Yao et al., 2022; Wu et al., 2024). In visual generation, agentic frameworks further demonstrate that specialized agents can cooperate for image generation, editing, story understanding, and narrative visualization (Wang et al., 2024; Kim et al., 2025; Zhou et al., 2025). Human-AI comic creation systems also show that intermediate authoring interfaces can improve creative control by allowing users to guide story structure, visual composition, or generated comic content (Chen and Jhala, 2024; Jin and Song, 2023). Existing agentic visual systems mainly target general image generation/editing or story visualization, rather than manga-specific end-to-end production. They lack explicit control over editable page layouts, reusable visual references, and cross-panel consistency. MangaFlow fills this gap with an agentic framework for controllable manga generation.

3 Methodology

3.1 Problem Definition

Given a story prompt x, a manga generation system aims to produce a complete manga:

$$\hat{M} = \{\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_N\}, \tag{1}$$

where each page $\hat{P}_i$ contains a page layout, a set of rendered panels, and optional text elements:

$$\hat{P}_i = (\hat{L}_i, \{\hat{I}_{i,j}\}_{j=1}^{n_i}, \{\hat{T}_{i,j}\}_{j=1}^{m_i}). \tag{2}$$

Here, $\hat{L}_i$ denotes the generated panel layout of page i, $\hat{I}_{i,j}$ denotes the j-th generated panel image, and $\hat{T}_{i,j}$ denotes a text element such as a speech bubble, narration box, thought bubble, or shout bubble.

In controllable manga generation, the system may also receive user constraints c, including the number of pages, target panel counts, manga style, language, user-defined layouts, reference manga layouts, character references, scene references, or key-object references. Formally, the generation process can be written as:

$$\hat{M} = G(x, c, R), \tag{3}$$

where R denotes optional visual references.

3.2 Challenges

To the best of our knowledge, MangaFlow is the first end-to-end agentic system for controllable manga generation. This setting introduces four key challenges.

C1: Narrative decomposition and long-range consistency. Manga stories involve multiple scenes, recurring characters, and key objects. Generating panels independently can cause character, scene, and object drift across panels and pages.

C2: Explicit layout controllability. Manga layout is a core storytelling structure rather than a purely aesthetic outcome. A practical system should follow user-defined panel counts and geometry while supporting reference-layout reuse.

C3: Reference-guided panel rendering. Users may provide or edit character sheets, scene references, layout, and key-object designs. The system should reuse these references across panels instead of relying solely on textual prompts.

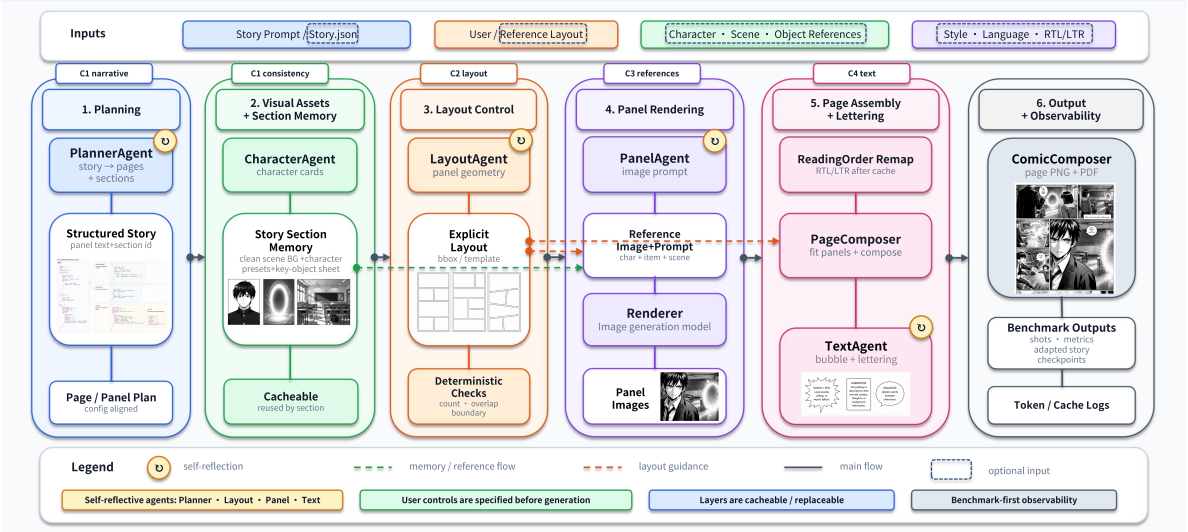


Figure 2: The MangaFlow framework decomposes manga generation into story planning, story-section memory construction, explicit layout control, reference-conditioned panel rendering, page composition, and lettering. MangaFlow can take either a simple story prompt as input or additional user-specified conditions, such as a detailed story JSON file, reference images for characters, scenes, and key objects, and target page layouts, to enable more precise and controllable manga generation.

C4: Page composition and lettering. A manga page is more than a set of independent images. Rendered panels must be composed according to target layout, while speech bubbles and narration boxes should preserve readability and narrative flow.

3.3 Framework Overview

MangaFlow addresses these challenges by decomposing manga creation into a structured agentic pipeline. The overall framework is shown in Figure 2. Given a story prompt, MangaFlow first plans the story and divides it into story sections. Each story section maintains its own memory, including textual descriptions and visual references for shared scenes, characters, and key objects. The layout agent then produces or retrieves page layouts, which are further corrected and refined. The panel agent constructs panel-level prompts using the story context, layout, and story-section memory. The renderer generates each panel image with reference conditioning. Finally, text agent adds text bubbles based on composed pages.

3.4 Story Planning and Story Section Memory

To address **C1**, MangaFlow introduces **Story Section Memory**. Instead of treating the story as a flat sequence of panels, the planner decomposes the input prompt into pages, panels, and story sections:

$$\mathcal{S} = \{s_1, s_2, \ldots, s_K\}. \tag{4}$$

Each story section s_k corresponds to a coherent narrative segment:

$$s_k = (d_k, e_k, C_k, O_k), \tag{5}$$

where d_k is the textual section description, e_k is the shared scene, C_k is the character set, and O_k is the set of key objects.

For each section, MangaFlow builds a memory:

$$\mathcal{M}_k = (d_k, R_k^{scene}, R_k^{char}, R_k^{obj}, \phi_k), \tag{6}$$

where R_k^{scene}, R_k^{char}, and R_k^{obj} denote scene, character, and object references (texts and images), and ϕ_k denotes structured profiles such as character presets and key-object descriptions. Panels in the same section reuse this memory to preserve scene, character, and object consistency.

For a panel (i, j) assigned to section $z(i, j)$, the panel prompt is constructed as:

$$p_{i,j} = A_{\text{panel}}(y_{i,j}, L_i, \mathcal{M}_{z(i,j)}), \tag{7}$$

where $y_{i,j}$ is the panel-level story description and L_i is the layout of page i. This design makes consistency an explicit reusable condition rather than an implicit property of long prompts.

3.5 Explicit Layout Control

To address **C2**, MangaFlow treats layout as a first-class structural variable. Each page layout is represented as a set of panel regions:

$$L_i = \{B_{i,1}, B_{i,2}, \ldots, B_{i,n_i}\}, \tag{8}$$

Table 1: Design rationale for the intermediate states of MangaFlow: the control factor each state exposes, its role in the pipeline, and the limitation of direct page generation that motivates it.

Intermediate state	Control factor	Role	Limitation of direct page generation
Story plan	Narrative decomposition	Assigns story events to pages and panels.	May skip, merge, or misplace events.
Story Section Memory	Long-range consistency	Reuses character, scene, and object information within coherent story sections.	Long prompts may cause character/scene/object drift.
Explicit layout	Panel geometry	Represents layouts as concrete panel regions for deterministic composition.	May produce wrong panel counts or distorted panel boundaries.
Visual references	Visual grounding	Provides reusable character, scene, and object references.	Text-only descriptions may not preserve fine-grained identity.
Lettering representation	Readability	Separates dialogue/narration placement from image rendering.	May generate missing, unreadable, or poorly placed text.

$B_{i,j}$ denotes a rectangular or polygonal region.

MangaFlow supports three layout sources: manual specification, reference-layout extraction from manga pages, and layout-agent generation:

$$L_i = A_{\text{layout}}(q_i, n_i, c), \tag{9}$$

where q_i is the page-level story context and n_i is the target panel number. The layout agent then performs a self-reflection step that jointly evaluates narrative appropriateness, spatial balance, and geometric validity. As its deterministic correction component, the layout is projected as:

$$\tilde{L}_i = \Pi(L_i), \tag{10}$$

where $\Pi(\cdot)$ checks panel count, overlap, page boundaries, blank regions, and layout constraints.

3.6 Reference-Conditioned Panel Rendering

To address **C3**, MangaFlow supports reference-conditioned rendering with explicit visual assets. Users may provide character, scene, and key-object references before generation; if absent, MangaFlow automatically constructs them through story-section memory. Reference manga pages can also be used as structural references by extracting their layouts, which are reused during page composition. For each panel, MangaFlow composes the references associated with its story section:

$$R_{i,j} = \text{ComposeRef}(R^{scene}_{z(i,j)}, R^{char}_{z(i,j)}, R^{obj}_{z(i,j)}), \tag{11}$$

where $z(i,j)$ denotes the story section assigned to panel (i,j). The renderer generates the panel image as:

$$\hat{I}_{i,j} = \mathcal{R}(p_{i,j}, R_{i,j}, w_{i,j}, h_{i,j}), \tag{12}$$

where $p_{i,j}$ is the panel prompt, and $w_{i,j}, h_{i,j}$ are determined by the target panel region. This design gives users explicit control over characters, scenes, objects, and layout structure.

3.7 Page Composition and Lettering

To address **C4**, MangaFlow first composes rendered panels into full pages according to the layout:

$$\hat{P}^{img}_i = \text{ComposePage}(\{\hat{I}_{i,j}\}_{j=1}^{n_i}, \tilde{L}_i). \tag{13}$$

The text agent then places speech bubbles, narration boxes, thought bubbles, and shout bubbles:

$$\hat{P}_i = A_{\text{text}}(\hat{P}^{img}_i, \{t_{i,j}\}, \tilde{L}_i). \tag{14}$$

Subject centers or speaker head boxes detected by multimodal LLM guide bubble placement. Finally, all pages are assembled into the complete manga:

$$\hat{M} = \text{ComposeComic}(\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_N). \tag{15}$$

3.8 Design Rationale for the Intermediate States

The intermediate states of MangaFlow are designed to expose manga-specific control factors that are entangled in direct page generation. As summarized in Table 1, each state addresses a distinct limitation of direct generation while providing an explicit and editable representation for downstream control. This motivates MangaFlow as a controllable intermediate-representation pipeline rather than a direct page-generation system.

4 MangaGen-MetaBench

We build **MangaGen-MetaBench** on ViStoryBench (Zhuang et al., 2025), which provides diverse story prompts, character settings, and evaluation dimensions for narrative-driven visual generation. However, ViStoryBench targets image-sequence story visualization rather than manga generation, which additionally requires page-level layout control, panel structure, page composition, and lettering. MangaGen-MetaBench therefore augments ViStoryBench with manga-specific constraints and metrics.

4.1 Benchmark Construction

For each story sample, we keep the original narrative prompt and character or style information from ViStoryBench when available. We then convert the story visualization setting into a manga generation setting by adding manga-specific page constraints. Each sample is associated with a page number, target panel counts and target layouts. The target layouts are generated by Layout agent. The resulting benchmark evaluates whether a method can transform a story prompt into a complete manga page sequence while respecting layout and consistency constraints. This benchmark should be understood as a *meta-benchmark* rather than a complete manga dataset. Its purpose is to reuse the story diversity and consistency evaluation foundation of ViStoryBench, while introducing additional manga-specific measurements.

4.2 Metrics

MangaGen-MetaBench evaluates two groups of abilities. We inherit ViStoryBench metrics to measure story-level visual consistency, narrative alignment, and generation quality, and introduce manga-specific metrics for page-level layout controllability, lettering placement, and readability. Since the inherited metrics are not our contribution, the metric details are provided in Appendix A.1.3.

Inherited Story Visualization Metrics. We adopt the original ViStoryBench metrics, including **CIDS** for character identity consistency, **CSD** for style consistency, **PA** for prompt alignment, and **OCCM** for character set correctness. We also report **Inc**, **Aes**, and **CP** to assess image quality, aesthetic quality, and copy-paste-style pseudo-consistency. Together, these metrics evaluate whether generated manga panels faithfully visualize the input story and remain visually coherent.

Layout Controllability. For page i, let $L_i = \{B_{i,k}\}_{k=1}^{|L_i|}$ be the target layout and $\hat{L}_i = \{\hat{B}_{i,j}\}_{j=1}^{|\hat{L}_i|}$ be the generated or detected layout, where each $B_{i,k}$ and $\hat{B}_{i,j}$ is a panel region. Let P_i denote the full page region and $A(\cdot)$ denote the area operator. We evaluate whether the generated manga follows specified page structure using following metrics.

Panel Count Accuracy evaluates whether the generated page contains the required number of panels:

$$\mathrm{Acc}_{count} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[|\hat{L}_i| = |L_i|\right]. \tag{16}$$

Layout IoU measures the geometric agreement between the generated and target panel layouts. For each page, we construct a one-to-one matching $\mathcal{M}_i$ between target and generated panels by greedily selecting panel pairs in descending IoU order. The score is averaged over target panels, so unmatched target panels contribute zero:

$$\mathrm{IoU}_{layout} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|L_i|} \sum_{(k,j)\in\mathcal{M}_i} \frac{A(B_{i,k} \cap \hat{B}_{i,j})}{A(B_{i,k} \cup \hat{B}_{i,j})}. \tag{17}$$

Coverage Ratio measures page-space utilization, defined as the fraction of the page covered by the union of generated panel regions:

$$\mathrm{Coverage} = \frac{1}{N}\sum_{i=1}^{N} \frac{A\left(\bigcup_{j=1}^{|\hat{L}_i|} \hat{B}_{i,j}\right)}{A(P_i)}. \tag{18}$$

Overlap Ratio measures structural invalidity caused by overlapping panels. It is computed as the fraction of page area covered by two or more generated panels:

$$\mathrm{Overlap} = \frac{1}{N}\sum_{i=1}^{N} \frac{A\left(\left\{p \in P_i : \sum_{j=1}^{|\hat{L}_i|} \mathbb{1}[p \in \hat{B}_{i,j}] \geq 2\right\}\right)}{A(P_i)}. \tag{19}$$

Lettering and Readability. We evaluate lettering and readability with two metrics.

Bubble Placement Score is evaluated by human annotators. For each panel containing text bubbles, annotators check whether any bubble occludes a character's face. A panel receives 0 if such face occlusion occurs, and 1 otherwise. The final score is normalized over all evaluated panels:

$$\mathrm{BPS} = \frac{1}{|\mathcal{P}^{txt}|} \sum_{p\in\mathcal{P}^{txt}} \mathbb{1}[\text{no face occlusion}(p)], \tag{20}$$

where $\mathcal{P}^{txt}$ denotes panels containing text bubbles.

Manga Readability Score is evaluated with a multimodal LLM. The model first reads the generated manga and summarizes the story, then compares the summary with the ground-truth story. The score is assigned on a 1–5 scale according to how well the generated manga communicates the original story. Detailed annotation and judging protocols are provided in Appendix A.2.

5 Experiments

5.1 Experimental Setup

MangaGen-MetaBench extends ViStoryBench (Zhuang et al., 2025), including 80

stories with manga-specific layout and lettering evaluations. The inherited metrics measure whether generated panels faithfully visualize the story, while our added metrics evaluate whether the outputs satisfy manga-specific requirements.

Compared Methods. We compare two categories of methods. The first category is direct page generation, where the same image generation backbones used in MangaFlow receive the full page description, including the target layout and panel information, and directly generate the whole manga page. This category tests whether current image generation models can follow explicit manga page specifications. Specifically, we use Gemini 2.5 Flash Image (Google) and FLUX.2 9B (Black Forest Labs) as direct generation baselines. The second category is MangaFlow with different renderers, where the same generation backbones are used as panel renderers within our structured pipeline.

Metrics. We report three metric groups: inherited ViStoryBench metrics for story visualization quality, layout metrics for target page-structure adherence, and human evaluation for bubble placement and LLM evaluation for readability. Details can be found in Appendix A.1.3

5.2 Main Results

Table 2: CIDS and CSD denote self character similarity and self style similarity, respectively. PA is the average prompt-alignment score. CM denotes OCCM, Inc denotes inception score, and Aes denotes aesthetic score. Please refer to Table. 8 for full metric results.

Method	CIDS↑	CSD↑	PA↑	CM↑	Inc↑	Aes↑
Direct-Gem.	0.562	0.702	2.708	51.60	8.14	4.52
Direct-FLUX	0.591	**0.750**	2.037	39.05	2.24	4.56
MF-Gem.	**0.643**	0.622	2.901	**73.93**	**12.26**	5.54
MF-FLUX	0.619	0.668	**3.299**	66.87	10.61	**5.92**

We compare direct page generation baselines with MangaFlow using different rendering backbones. Table 2 reports inherited ViStoryBench metrics, while Table 3 reports manga-specific metrics for layout controllability and readability. On ViStoryBench, MangaFlow achieves better performance than direct generation baselines on most metrics. Although Direct-FLUX obtains the highest self-style consistency, its weaker performance on other metrics suggests that direct page generation may preserve local style but still struggles with story alignment and generation quality.

Table 3: Manga-specific metrics. Layout metrics evaluate panel count, layout geometry, page coverage, and panel overlap. Bubble denotes a human-evaluated score for face-occlusion-free bubble placement, while Read. denotes a GPT-4.1-based readability score. Bubble scores are left blank for direct baselines because Direct-FLUX.2 often produces unreadable text, while Direct-Gemini frequently omits required dialogue.

System	Renderer	Layout Metrics				Text Metrics	
		Count ↑	IoU ↑	Cov. ↑	Ovl. ↓	Bubble ↑	Read. ↑
Direct	Gemini	27.94%	42.77%	88.70%	N/A	N/A	3.80
Direct	FLUX.2	44.20%	41.11%	73.65%	N/A	N/A	2.67
MangaFlow	Gemini	**100%**†	**100%**†	**99.98%**	0.62%	**97.4%**	**4.87**
	FLUX.2					97.1%	4.71

† For MangaFlow, the marked layout scores are deterministic due to explicit page composition. The two MangaFlow variants share the same explicit layouts and deterministic page composition, so their layout metrics are identical; only the rendering backbone differs. For direct baselines, Ovl. is marked as N/A because they do not generate explicit panel layouts, making panel overlap not directly applicable.

Table 3 shows that explicit page composition is crucial for controllable manga generation. Direct baselines struggle to recover panel counts and layout geometry from prompts alone, whereas MangaFlow follows the target page structure through deterministic composition. MangaFlow also achieves high page coverage and supports readability evaluation, while direct baselines often fail to produce reliable dialogue or bubble structures. These results show that MangaFlow's structured decomposition improves layout controllability and readability without sacrificing story visualization quality. Although the main experiments use ViStoryBench samples with detailed story and character specifications, MangaFlow also supports a simpler text-to-manga setting, where plans, layouts, references, and pages are generated automatically from a natural-language prompt. A qualitative example is shown in Fig. 1, with details in Appendix A.3.3.

Statistical significance and human validation. We further validate the main results with paired story-level statistical tests (Appendix A.3.4) and a blind human study of readability (Appendix A.3.5). The representative improvements remain significant after multiple-comparison correction. While human annotators are stricter than GPT-4.1 on absolute readability scores, they strongly agree with its relative comparison between methods (96.0% no-conflict agreement), and prefer MangaFlow in $24/25$ paired stories.

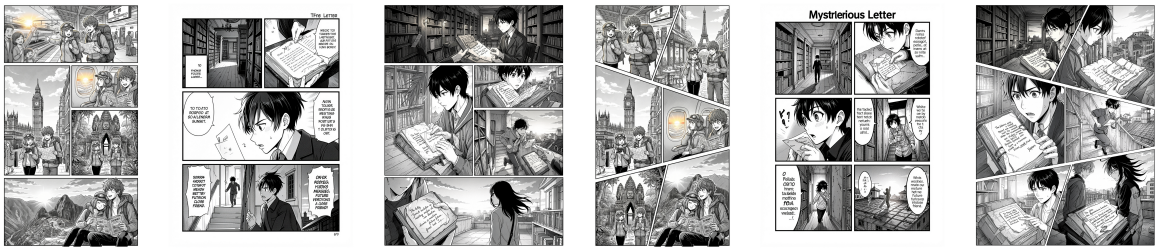


Figure 3: Visualization of reference-layout-guided manga generation. Each example contains a reference manga layout, a direct generation result, and the MangaFlow result. Direct page generation may fail to preserve the reference panel structure even when reference image is provided in the prompt, while MangaFlow explicitly reuses the extracted layout and generates new manga pages with more faithful panel arrangements.

5.3 Ablation Study

We conduct two ablation studies to analyze the contribution of key components in MangaFlow. First, we evaluate **Story Section Memory**. This ablation is evaluated with ViStoryBench consistency and quality metrics. Second, we evaluate **Layout Self-Reflection**. Since this component directly affects page structure, we evaluate it using page-space coverage and panel overlap. For each ablation, we report only the metrics that are directly related to the removed component, avoiding unrelated comparisons such as evaluating story-section memory with layout IoU.

Table 4: Ablation study based on FLUX.2 render on Story Section Memory. Please refer to Table. 9 for full metric results. CSD and CIDS are self-consistency scores. PA denotes prompt alignment, where Scene, Shot, CI, and IA correspond to scene alignment, camera/shot alignment, character-action alignment, and individual-action alignment, respectively.

Variant	CSD↑	CIDS↑	Scene↑	Shot↑	CI↑	IA↑	Aes↑
w/o Sec. Mem.	0.547	0.582	3.836	**3.176**	3.439	2.525	5.83
Full MF-FLUX	**0.668**	**0.619**	**3.861**	3.008	**3.638**	**2.688**	**5.92**

Story Section Memory. We remove Story Section Memory and keep the rest of the pipeline unchanged. Table 4 shows that the full MangaFlow-FLUX model improves both self-consistency metrics, including CSD and CIDS, compared with the variant without memory. It also achieves better results on most prompt-alignment dimensions, including Scene, CI, and IA, as well as a higher aesthetic score. This suggests that Story Section Memory helps maintain recurring characters and visual context across panels while supporting panel-level semantic alignment.

Table 5: Ablation on layout self-reflection. We evaluate its effect using page-space coverage and panel overlap.

Variant	Cov. ↑	Ovl. ↓
w/o Layout SR	98.78%	0.86%
Full MangaFlow	**99.98%**	**0.62%**

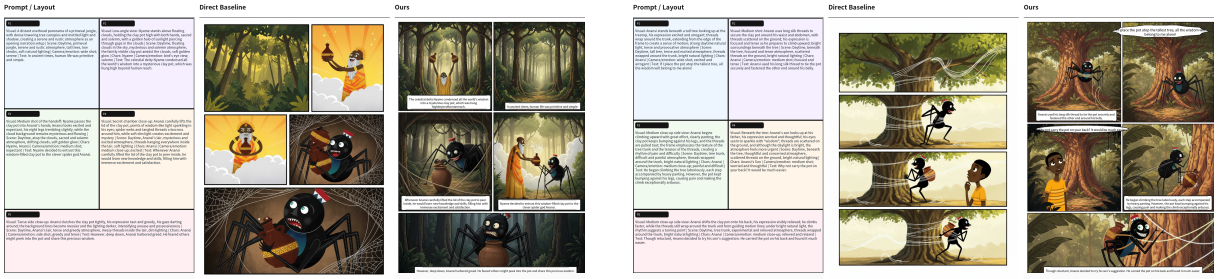
Layout Self-Reflection. Table 5 evaluates layout self-reflection. Removing it lowers page coverage and increases panel overlap, showing that this refinement step improves page-space utilization and produces cleaner panel geometry. We therefore keep layout self-reflection in MangaFlow and provide qualitative examples in Appendix A.3.6.

5.4 Visualization

Reference-layout-guided Generation. We further visualize whether different methods can follow layouts extracted from reference manga. In Figure 3, although the direct method receives the same reference manga layout information, it may fail to preserve the exact panel structure. In contrast, MangaFlow explicitly reuses the extracted layout as a structural constraint and generates new manga pages with more faithful panel arrangements.

Layout-controlled Generation. We visualize the difference between direct page generation and MangaFlow under the same story prompt and target layout. The comparison in Figure 4 highlights that direct generation methods fail to preserve the required panel count or panel geometry, while MangaFlow follows the explicit layout by construction.

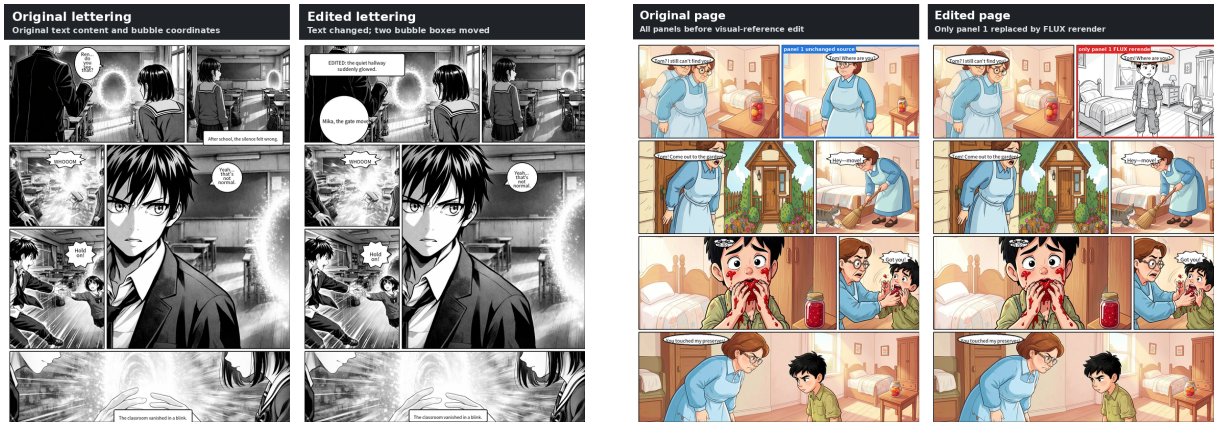
Text-placement editing. Text bubbles and narration boxes are placed according to explicit coordinates and text specifications. If the bubble position or text content is edited from T to T', the lettering



(a) Example 1

(b) Example 2

Figure 4: Qualitative comparison of layout-controlled manga generation. For each example, the left side visualizes the specified target layout together with the prompt assigned to each panel, the middle shows the result of a direct page generation method, and the right side shows the result of MangaFlow. Although the direct method receives the same layout and panel-level prompt information, it may fail to preserve the required panel count or panel geometry, while MangaFlow follows the explicit layout through deterministic page composition.



(a) Text-placement editing

(b) Visual-reference editing

Figure 5: Qualitative examples of explicit editing in MangaFlow. (a) Text placement and content can be directly modified through the lettering specification and are deterministically rendered by the compositor. (b) Visual references can be replaced explicitly, and the updated reference is directly propagated to the panel renderer.

component directly uses T', and the edited bubbles and text are rendered at the specified coordinates by the same compositor. The edited text placement is thus deterministically executed (Fig. 5(a)).

Visual-reference editing. Visual references are explicit inputs to the panel renderer. To visualize the editing capability, we replace the reference image used by one selected panel. The corresponding panel is re-rendered with the new reference, whereas the remaining panels are preserved (Fig. 5(b)).

6 Conclusion

We introduced **MangaFlow**, an end-to-end agentic framework for language-guided and layout-controllable manga generation. MangaFlow treats manga creation as a structured generation process that coordinates story planning, section-level memory, visual references, explicit layouts, panel rendering, page composition, and lettering. This formulation provides a controllable generation paradigm for long-form manga generation. By making layout and references explicit intermediate variables, MangaFlow supports both simple text-to-manga generation and more precise generation with user-specified layouts or visual assets.

We further proposed **MangaGen-MetaBench**, a meta-benchmark for evaluating manga generation beyond image-sequence story visualization. By incorporating manga-specific dimensions such as layout adherence, page composition, lettering, and readability, MangaGen-MetaBench provides an initial evaluation protocol for controllable manga generation. We hope MangaFlow and MangaGen-MetaBench can serve as a step toward more structured, controllable, and evaluable systems for long-form manga generation.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP23K28139 and JP25H01149.

Limitations

MangaFlow demonstrates that controllable manga generation benefits from explicit intermediate structures, but several limitations remain. First, the final visual quality still depends on the underlying image generation backbone, so failures in character rendering, fine-grained actions, or local visual details may affect the generated panels. Second, although MangaFlow introduces explicit layout control and lettering, speaker-aware bubble placement remains challenging in complex panels, especially when stylized manga faces are difficult to localize. Third, MangaGen-MetaBench is designed as a meta-benchmark built on existing story visualization data, rather than a fully annotated manga dataset with human-designed layouts, bubble positions, and speaker annotations.

Ethical Considerations

Manga generation systems may raise concerns about copyright, artist consent, and style imitation, especially when user-provided references resemble the work of specific artists. In practical use, reference images should be licensed or user-owned, and generated outputs should not be presented as human-created artwork without disclosure. The system may also generate inappropriate or biased visual content depending on the underlying generator and input story, so content filtering and provenance mechanisms are important for deployment.

References

Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, and Hikaru Ikuta. 2020. Building a manga dataset "manga109" with annotations for multimedia applications. *IEEE multimedia*, 27(2):8–18.

Black Forest Labs. FLUX.2 [klein]. https://huggingface.co/black-forest-labs/FLUX.2-klein-9b-kv. Hugging Face model card. Accessed: 2026-05-03.

Ying Cao, Antoni B Chan, and Rynson WH Lau. 2012. Automatic stylistic manga layout. *ACM Transactions on Graphics (TOG)*, 31(6):1–10.

Hong Chen, Rujun Han, Te-Lin Wu, Hideki Nakayama, and Nanyun Peng. 2022. Character-centric story visualization via visual planning and token alignment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8259–8272.

Siyu Chen, Dengjie Li, Zenghao Bao, Yao Zhou, Lingfeng Tan, Yujie Zhong, and Zheng Zhao. 2024. Manga generation via layout-controllable diffusion. *arXiv preprint arXiv:2412.19303*.

Yi-Chun Chen and Arnav Jhala. 2024. Collaborative comic generation: Integrating visual narrative theories with ai models for enhanced creativity. *arXiv preprint arXiv:2409.17263*.

Siyuan Feng, Teruya Yoshinaga, Katsuhiko Hayashi, Koki Washio, and Hidetaka Kamigaito. 2024. How panel layouts define manga: Insights from visual ablation experiments. *arXiv preprint arXiv:2412.19141*.

Silin Gao, Sheryl Mathew, Li Mi, Sepideh Mamooler, Mengjie Zhao, Hiromi Wakaki, Yuki Mitsufuji, Syrielle Montariol, and Antoine Bosselut. 2025. Vinabench: Benchmark for faithful and consistent visual narratives. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2870–2879.

Google. Gemini 2.5 Flash Image. https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-image. Google AI for Developers documentation. Accessed: 2026-05-03.

Huiguo He, Huan Yang, Zixi Tuo, Yuan Zhou, Qiuyue Wang, Yuhang Zhang, Zeyu Liu, Wenhao Huang, Hongyang Chao, and Jian Yin. 2025. Dreamstory: Open-domain story visualization by llm-guided multi-subject consistent diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2023. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176.

Ze Jin and Zorina Song. 2023. Generating coherent comic with rich story using chatgpt and stable diffusion. *arXiv preprint arXiv:2305.11067*.

Guangmei Jing, Yongtao Hu, Yanwen Guo, Yizhou Yu, and Wenping Wang. 2015. Content-aware video2comics with manga-style layout. *IEEE Transactions on Multimedia*, 17(12):2122–2133.

Seungkwon Kim, GyuTae Park, Sangyeon Kim, and Seung-Hun Nam. 2025. Visagent: Narrative-preserving story visualization framework. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yingxuan Li, Kiyoharu Aizawa, and Yusuke Matsui. 2024. Manga109dialog: A large-scale dialogue dataset for comics speaker detection. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6329–6338.

Adyasha Maharana, Darryl Hannan, and Mohit Bansal. 2022. Storydall-e: Adapting pretrained text-to-image transformers for story continuation. In *European conference on computer vision*, pages 70–87. Springer.

Jiawei Mao, Xiaoke Huang, Yunfei Xie, Yuanqi Chang, Mude Hui, Bingjie Xu, Zeyu Zheng, Zirui Wang, Cihang Xie, and Yuyin Zhou. Story-iter: A training-free iterative paradigm for long story visualization. In *The Fourteenth International Conference on Learning Representations*.

Chutian Meng, Fan Ma, Chi Zhang, Jiaxu Miao, Yi Yang, and Yueting Zhuang. Logistory: A logic-aware framework for multi-image story visualization. In *The Fourteenth International Conference on Learning Representations*.

Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. 2023. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 643–654.

Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. 2023. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2493–2502.

Xiaoqian Shen and Mohamed Elhoseiny. 2025. Storygpt-v: Large language models as consistent story visualizers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13273–13283.

Emanuele Vivoli, Marco Bertini, and Dimosthenis Karatzas. 2024a. Comix: A comprehensive benchmark for multi-task comic understanding. *Advances in Neural Information Processing Systems*, 37:140828–140846.

Emanuele Vivoli, Mohamed Ali Souibgui, Andrey Barsky, Artemis LLabres, Marco Bertini, and Dimosthenis Karatzas. 2024b. One missing piece in vision and language: A survey on comics understanding. *arXiv preprint arXiv:2409.09502*.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. 2024. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395.

Jianzong Wu, Chao Tang, Jingbo Wang, Yanhong Zeng, Xiangtai Li, and Yunhai Tong. 2025. Diffsensei: Bridging multi-modal llms and diffusion models for customized manga generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28684–28693.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. 2023. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7452–7461.

Xin Yang, Zongliang Ma, Letian Yu, Ying Cao, Baocai Yin, Xiaopeng Wei, Qiang Zhang, and Rynson WH Lau. 2021. Automatic comic generation with stylistic multi-page layouts and emotion-driven text balloon generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2):1–19.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Jinlu Zhang, Jiji Tang, Rongsheng Zhang, Tangjie Lv, and Xiaoshuai Sun. 2025. Storyweaver: A unified world model for knowledge-enhanced story character customization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9951–9959.

Tianchen Zhou, Zhongjie Duan, Cen Chen, Wenmeng Zhou, Yanhao Wang, and Yaliang Li. 2025. Agentstory: A multi-agent system for story visualization with multi-subject consistent text-to-image generation. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pages 1894–1902.

Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. 2024. Storydiffusion: Consistent self-attention for long-range image and video generation. *Advances in Neural Information Processing Systems*, 37:110315–110340.

Cailin Zhuang, Ailin Huang, Yaoqi Hu, Jingwei Wu, Wei Cheng, Jiaqi Liao, Hongyuan Wang, Xinyao Liao, Weiwei Cai, Hengyuan Xu, et al. 2025. Vistorybench: Comprehensive benchmark suite for story visualization. *arXiv preprint arXiv:2505.24862*.

A Appendix

A.1 Experiment and Method details

For all external models and datasets, we follow their original terms of use and cite the corresponding creators or model providers. We use these artifacts only for research evaluation. The LLM used for our agents is GPT 5.4-nano.

Due to copyright-protection constraints of the Nano-Banana image generation API, a subset of ViStoryBench stories could not be completed with this renderer. As a result, all Nano-Banana-related results are computed on the 62 stories for which the API returned valid outputs. For a fair comparison, the corresponding direct Nano-Banana baseline and MangaFlow-Nano-Banana variant are evaluated on the same subset.

A.1.1 Statistics of Experiment Result

We report aggregate metrics over the full evaluation set and clarify that automatic results are from one complete generation run per method, not the maximum over multiple trials. For human evaluation, we report mean scores across annotators and samples.

A.1.2 Computational Budget

We evaluate 80 stories in MangaGen-MetaBench. FLUX.2 9B experiments are conducted using local inference with a single NVIDIA A6000 Ada GPU. The total computational budget for FLUX.2-based generation is approximately 24 GPU hours.

A.1.3 Vistory Metrics

To comprehensively evaluate the performance of story generation models, we adopt the evaluation framework introduced by ViStoryBench(Zhuang et al., 2025), which comprises a suite of multi-dimensional automated metrics across six key perspectives.

Character Identification Similarity (CIDS): This metric evaluates identity consistency by computing the cosine similarity between character features. It calculates both *Cross-CIDS* (similarity between generated images and reference images) and *Self-CIDS* (consistency across generated frames). The pipeline utilizes Grounding DINO for character detection, followed by feature extraction using FaceNet/ArcFace for realistic characters and CLIP ViT-L/14 for stylized characters.

Style Similarity (CSD): To measure stylistic coherence independent of semantic content, this metric utilizes a style-trained CLIP vision encoder combined with Content-Style Disentanglement (CSD) layers. It evaluates both *Cross-CSD* and *Self-CSD* via pairwise cosine similarity of the extracted style embeddings.

Prompt Alignment: Visual-Language Models (VLMs) are employed as automated evaluators to score the semantic alignment between generated images and text prompts on a 0-4 Likert scale. This is sub-divided into four fine-grained aspects: *Scene Score* (environment and background fidelity), *Shot Score* (camera perspective and composition), *Character Interaction* (relational dynamics between characters), and *Individual Action* (accuracy of specific gestures and poses). In our paper, we use GPT4.1 as automated evaluator.

Onstage Character Count Matching (OCCM): This metric quantifies the model's ability to generate the correct number of characters as specified in the script. It applies a non-linear exponential penalty to the relative error between the detected character count D and the expected count E:

$$OCCM = \exp\left(-\frac{|D-E|}{\epsilon+E}\right) \times 100\%$$

Copy-Paste Detection: To penalize models that bypass true generation by simply replicating input references, this metric calculates a Softmax-based Copy-Paste Rate. It assesses whether a generated character feature is excessively close to a specific anchor reference in the latent space, indicating overfitting rather than generalized learning.

Aesthetic Quality and Diversity: Finally, generation quality is assessed using the *Inception Score (IS)* to measure visual diversity and clarity, along with an *Aesthetic Score* evaluated via an aesthetic predictor to quantify the overall artistic appeal and perceptual quality of the generated sequence.

Table 6: Compared methods. "Direct" means generating the whole manga page in one step. "Pipeline" means generating panels through MangaFlow and composing them with explicit layouts.

Method	Type	Layout	Sec. Mem.
Direct-Gemini	Direct	Prompt	–
Direct-FLUX.2	Direct	Prompt	–
MangaFlow-Gemini	Pipeline	Explicit	✓
MangaFlow-FLUX.2	Pipeline	Explicit	✓

Table 7: Metric groups in MangaGen-MetaBench. ViStoryBench metrics are inherited, while layout and lettering/readability metrics are added for manga generation.

Group	Metrics
ViStoryBench	CIDS, CSD, PA, OCCM, Inc, Aes, CP
Layout	Count Acc., Layout IoU, Coverage, Overlap
Human and LLM	Bubble Placement, Readability

A.2 Details of Lettering and Readability Evaluation

A.2.1 Bubble Placement Annotation

We use human annotation to evaluate bubble placement because automatic methods cannot reliably determine whether a speech bubble occludes an important facial region in stylized manga images. Annotators evaluate each panel containing text bubbles. A panel is assigned a binary score:

$$s_p = \begin{cases} 0, & \text{if any bubble occludes a character's face,} \\ 1, & \text{otherwise.} \end{cases} \tag{21}$$

The final Bubble Placement Score is the average over all panels containing text bubbles:

$$\mathrm{BPS} = \frac{1}{|\mathcal{P}^{txt}|} \sum_{p \in \mathcal{P}^{txt}} s_p. \tag{22}$$

We only penalize face occlusion caused by text bubbles. Bubbles crossing panel boundaries or gutters are allowed as long as they do not damage the readability of the page or occlude character faces.

A.2.2 Human Annotation for Bubble Placement

We use human annotation only for evaluating bubble placement quality. The annotation task does not involve collecting personal information from annotators or evaluating human subjects. Annotators are asked to inspect generated manga panels that contain text bubbles and determine whether any bubble occludes a character's face.

We recruited 3 annotators for this evaluation. Each panel containing text bubbles was evaluated by 3 annotators, and the final Bubble Placement Score is computed by averaging scores over annotators and evaluated panels.

The annotation instruction given to annotators was:

> Please inspect each manga panel containing one or more text bubbles. Assign a score of 1 if no text bubble occludes a character's face. Assign a score of 0 if any text bubble covers or substantially occludes a character's face. Only penalize face occlusion caused by text bubbles. Bubbles crossing panel boundaries or gutters should not be penalized unless they damage readability or occlude character faces.

Annotators were informed about the purpose of the annotation task and gave consent to participate. They received no monetary compensation. The task only requires annotators to judge generated images and does not ask for any sensitive personal information. If multiple annotators evaluated the same panel, we report the mean score across annotators and samples. In cases of disagreement, we use the average score. The annotation results are used only for aggregate evaluation and are not used to train any model.

Algorithm 1 MangaFlow Generation Pipeline

Require: Story prompt x, configuration c, optional layout L^{user}, optional references R^{user}, optional template library $\mathcal{T}$
Ensure: Generated manga $\hat{M}$

$Y \leftarrow A_{\text{planner}}(x, c)$
Enforce configured page number and target panel counts in Y
Extract story sections $\mathcal{S} = \{s_k\}_{k=1}^{K}$
for each story section $s_k \in \mathcal{S}$ **do**
  Load cached section memory $\mathcal{M}_k$ if available
  if $\mathcal{M}_k$ is unavailable **then**
    Generate or assign scene, character, and key-object references
    Build $\mathcal{M}_k = (d_k, R_k^{scene}, R_k^{char}, R_k^{obj}, \phi_k)$
  end if
end for
for each page $i = 1, \ldots, N$ **do**
  if user layout L_i^{user} is provided **then**
    $L_i \leftarrow L_i^{user}$
  else if a matched template exists in $\mathcal{T}$ **then**
    $L_i \leftarrow \mathrm{RetrieveTemplate}(q_i, n_i, \mathcal{T})$
  else
    $L_i \leftarrow A_{\text{layout}}(q_i, n_i, c)$
  end if
  $\tilde{L}_i \leftarrow \Pi(L_i)$
  for each panel $j = 1, \ldots, n_i$ **do**
    $k \leftarrow z(i, j)$
    $p_{i,j} \leftarrow A_{\text{panel}}(y_{i,j}, \tilde{L}_i, \mathcal{M}_k)$
    $R_{i,j} \leftarrow \mathrm{ComposeRef}(\mathcal{M}_k, R^{user})$
    $\hat{I}_{i,j} \leftarrow \mathcal{R}(p_{i,j}, R_{i,j}, w_{i,j}, h_{i,j})$
  end for
  $P_i^{img} \leftarrow \mathrm{ComposePage}(\{\hat{I}_{i,j}\}, \tilde{L}_i)$
  $\hat{P}_i \leftarrow A_{\text{text}}(P_i^{img}, Y, \tilde{L}_i)$
end for
$\hat{M} \leftarrow \mathrm{ComposeComic}(\{\hat{P}_i\}_{i=1}^{N})$
return $\hat{M}$

A.2.3 LLM-based Manga Readability Evaluation

We evaluate manga readability using a multimodal LLM judge. Instead of asking the judge for an overall preference score, we evaluate whether the generated manga can faithfully communicate the intended story. The evaluation is performed in two steps. First, the judge reads the generated manga pages and writes a concise summary of the perceived story. Second, the judge compares this generated summary with the ground-truth story and assigns a readability score from 1 to 5.

The scoring rubric is as follows:

- **1**: The generated manga is largely unreadable or fails to communicate the ground-truth story.
- **2**: The manga conveys only a few isolated story elements, but the main plot is missing or incorrect.
- **3**: The manga conveys the rough story direction, but important events, character actions, or causal relations are missing.
- **4**: The manga communicates most of the ground-truth story with minor omissions or ambiguities.
- **5**: The manga clearly and faithfully communicates the ground-truth story.

We use the following prompt template for the multimodal LLM judge:

```
You are evaluating the readability of a generated manga.

First, carefully read the provided manga pages and summarize
      the story conveyed by the manga.
Then compare your summary with the ground-truth story.
Assign a score from 1 to 5 according to how faithfully the
      manga communicates the ground-truth story.

Use a strict story-overlap criterion, not a generous
      impression score.
Estimate how much of the ground-truth story is communicated by
       the manga as an overlap percentage from 0 to 100.
Base this overlap on the ground-truth story elements that a
      reader can actually infer from the pages:
- main characters and their identities/roles;
- key settings and setting transitions;
- important events and actions;
- causal relations between events;
- temporal/order relations;
- ending state or resolution.

Important judging rules:
- Ambiguous, unreadable, or visually unclear content counts as
       missing.
- Incorrectly depicted events count as missing, even if the
      image is visually plausible.
- Extra unrelated events do not compensate for missing ground-
      truth events.
- Do not reward image quality, style, or aesthetics unless
      they directly improve story communication.
- If you are uncertain between two scores, choose the lower
      score.
- The score must exactly follow the overlap thresholds below.

Scoring rubric:
1: overlap < 40%. The manga is largely unreadable or fails to
      communicate the ground-truth story.
2: 40% <= overlap < 60%. The manga conveys some isolated or
      partial story elements, but the main plot is
      substantially missing or incorrect.
3: 60% <= overlap < 80%. The manga conveys the rough story
      direction, but important events, character actions,
      settings, or causal/order relations are missing.
4: 80% <= overlap < 90%. The manga communicates most of the
      ground-truth story, with only moderate omissions or
      ambiguities.
5: overlap >= 90%. The manga clearly and faithfully
      communicates almost all of the ground-truth story; only
      very minor details may be missing.

Ground-truth story:
{ground_truth_story}

Return your answer as strict JSON with this schema:
{
  "summary": "...",
  "overlap_percent": integer from 0 to 100,
  "matched_elements": ["..."],
  "missing_or_wrong_elements": ["..."],
  "score": 1,
  "reason": "..."
}
```

A.3 Additional Experiment Analysis

A.3.1 Story Section Memory Ablation Visualization

We provide qualitative examples to illustrate the effect of Story Section Memory. The visualization shows that reusing section-level character and scene references helps MangaFlow maintain more stable visual identities across panels and pages. As shown in Figure 6, recurring characters, background scenes, and important objects remain more consistent throughout the generated manga.

A.3.2 Renderer Visualization

We provide additional qualitative examples as shown in Fig.7 to compare different rendering backbones within the same MangaFlow pipeline. The story plan, story-section memory, layout, and page composition are kept unchanged, while only the panel renderer is changed. This visualization helps isolate the effect of the rendering backbone on visual appearance, consistency, and panel-level image quality.

A.3.3 Text-to-Manga Generation from a Simple Prompt

To further demonstrate the end-to-end generation capability of MangaFlow, we provide an example of text-to-manga generation from a simple story prompt. Unlike settings that require detailed storyboards, predefined layouts, or manually provided visual references, this example only uses a short textual prompt as input. MangaFlow automatically decomposes the prompt into a structured story plan,

Table 8: Quantitative results on ViStoryBench. We compare direct page generation baselines with MangaFlow using different rendering backbones. CSD denotes style similarity, CIDS denotes character identity similarity, and PA denotes prompt alignment. For PA, Scene, Shot, CI, and IA denote scene alignment, camera/shot alignment, character-action alignment, and individual-action alignment, respectively. CM denotes OCCM, Inc denotes inception score, Aes denotes aesthetic score, and CP denotes copy-paste score.

Method	Model	CSD↑		CIDS↑		PA↑					CM↑	Inc↑	Aes↑	CP↓
		Cross	Self	Cross	Self	Scene	Shot	CI	IA	Avg.				
Direct	Gemini	0.373	0.702	0.454	0.562	3.253	2.762	2.846	1.971	2.708	51.60	8.14	4.52	0.493
Direct	FLUX.2	0.229	**0.750**	0.358	0.591	2.280	2.620	1.885	1.363	2.037	39.05	2.24	4.56	0.493
MangaFlow	Gemini	**0.383**	0.622	**0.589**	**0.643**	3.061	2.912	3.163	2.468	2.901	**73.93**	**12.26**	5.54	0.506
MangaFlow	FLUX.2	0.328	0.668	0.447	0.619	**3.861**	**3.008**	**3.638**	**2.688**	**3.299**	66.87	10.61	**5.92**	**0.483**

Table 9: Ablation study on Story Section Memory on ViStoryBench. We compare the full MangaFlow-FLUX pipeline with the variant without Story Section Memory. CSD denotes style similarity, CIDS denotes character identity similarity, and PA denotes prompt alignment. For PA, Scene, Shot, CI, and IA denote scene alignment, camera/shot alignment, character-action alignment, and individual-action alignment, respectively. CM denotes OCCM, Inc denotes inception score, Aes denotes aesthetic score, and CP denotes copy-paste score.

Variant	CSD↑		CIDS↑		PA↑					CM↑	Inc↑	Aes↑	CP↓
	Cross	Self	Cross	Self	Scene	Shot	CI	IA	Avg.				
w/o Sec. Mem.	0.327	0.547	0.442	0.582	3.836	**3.176**	3.439	2.525	3.244	**69.41**	**12.91**	5.83	0.503
Full MangaFlow-FLUX	**0.328**	**0.668**	**0.447**	**0.619**	**3.861**	3.008	**3.638**	**2.688**	**3.299**	66.87	10.61	**5.92**	**0.483**

constructs story sections, generates page layouts, renders individual panels, composes manga pages, and places dialogue text.

Figure 1 shows the generated four-page manga. The result demonstrates that MangaFlow can function as a complete text-to-manga system: starting from a natural-language prompt, it produces multi-page manga with coherent story progression, explicit panel structure, page composition, and lettering. This ability complements the controllable setting discussed in the main paper, where users may additionally provide story JSON files, reference images, or target layouts for more precise control.

The story prompt of Fig.1 is shown below:

```
After school, a boy named Ren and a girl named Mika enter a
      quiet classroom. As soon as they step inside, a
      mysterious light fills the room, and the classroom
      transforms into a magical gateway. They are suddenly
      transported to China, where they walk through crowded
      streets, ancient buildings, red lanterns, and food
      stalls. Before they can react, the scene changes to the
      United States, with tall skyscrapers, wide avenues, and
      bright city lights. Then they travel to Japan,
      surrounded by cherry blossoms, traditional shrines, and
      modern city streets. Finally, they appear in France,
      standing near elegant streets, art museums, and the
      Eiffel Tower.
Suddenly, the sky and scenery begin to twist and fade. The boy
       and girl are pulled back through the glowing portal and
       return to the classroom. They wake up at their desks
      with a shock, breathing quickly and looking around in
      confusion. The classroom is quiet again, but both of
      them remember the same impossible journey. They stare at
       each other, unsure whether it was only a dream or a
      real adventure.
```

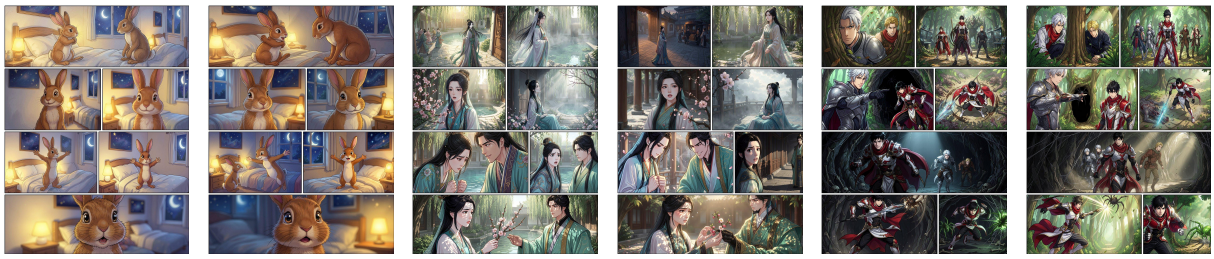
A.3.4 Statistical Significance of the Paired Story-Level Comparison

We further conduct paired story-level significance tests, treating each story as the sampling unit. We report 95% confidence intervals from $1{,}000$ bootstrap resamples and use two-sided Wilcoxon signed-rank tests with Holm correction; paired t-tests are used as a sensitivity analysis. As shown in Table 10, the representative improvements consistently exclude zero and remain significant after multiple-comparison correction.

A.3.5 Human–LLM Calibration Study for Readability

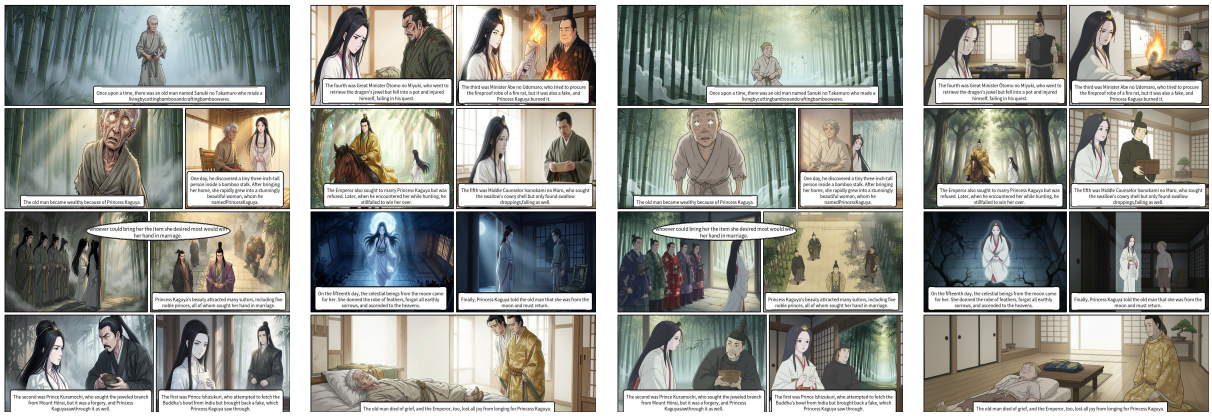
To validate the GPT-4.1 readability metric, we conduct a blind human study on 50 samples from 25 paired MangaFlow-Gemini and Direct-Gemini stories. Three annotators independently score each sample using the same 1–5 readability rubric as GPT-4.1.

As shown in Table 11, human annotators are systematically stricter than GPT-4.1 on the absolute scale, with a mean human-minus-LLM difference of -1.25. However, the relative method comparison is highly consistent: humans prefer MangaFlow in $24/25$ stories, while GPT-4.1 never prefers Direct. The human–LLM no-conflict agreement is 96.0%. We therefore use GPT-4.1 readability primarily as a relative comparison between methods, rather than as a calibrated absolute readability score.



(a) Example 1 ours (b) Example 1 w/o memory (c) Example 2 ours (d) Example 2 w/o memory (e) Example 3 ours (f) Example 3 w/o memory

Figure 6: Visualization of Story Section Memory. The pages show that MangaFlow can maintain consistent recurring characters, scenes, and key objects across panels and pages by reusing section-level textual and visual references.



FLUX.2, Page 1 FLUX.2, Page 2 Gemini, Page 1 Gemini, Page 2

Figure 7: Renderer comparison within MangaFlow. The same story plan, story-section memory, layout, and page composition are used, while the renderer is changed between FLUX.2 and Gemini 2.5 flash image. This comparison visualizes how different image generation backbones affect the final manga appearance under the same structured pipeline.

Table 10: Paired story-level improvement of MangaFlow over the Direct baseline, one representative metric per evaluation dimension. Confidence intervals are estimated from 1,000 bootstrap resamples, and p-values are from two-sided Wilcoxon signed-rank tests with Holm correction over all 34 comparisons.

Renderer	**Dimension (metric)**	**Δ (MangaFlow $-$ Direct) [95% CI]**	**Holm p**
FLUX	Character consistency (CIDS cross)	$+0.0893[0.0752, 0.1031]$	1.06×10^{-12}
FLUX	Layout adherence (IoU)	$+0.5820$ $[0.5580, 0.6046]$	2.59×10^{-13}
FLUX	Character–action alignment	$+1.7528[1.6319, 1.8712]$	3.32×10^{-13}
FLUX	Readability	$+2.0380$ $[1.8731, 2.2025]$	1.23×10^{-13}
Gemini	Character consistency (CIDS cross)	$+0.1391$ $[0.1037, 0.1779]$	8.54×10^{-8}
Gemini	Layout adherence (IoU)	$+0.5684$ $[0.5368, 0.5993]$	1.67×10^{-10}
Gemini	Character–action alignment	$+0.3121$ $[0.1978, 0.4324]$	6.53×10^{-5}
Gemini	Readability	$+0.9355$ $[0.7258, 1.1613]$	3.73×10^{-7}

A.3.6 Visualization of Self-Reflection

To better understand the effect of self-reflection in MangaFlow, we provide qualitative visualizations of the layout changes introduced by the Layout Agent. Each example in Fig. 8 contains two pages: the left page shows the original layout before self-reflection, while the right page shows the reflected layout after self-reflection. The visualization demonstrates how self-reflection helps reorganize panel allocation, reduce layout imbalance, and improve page-space usage without changing the underlying story content.

Table 11: Human–LLM calibration study for readability (50 samples from 25 paired stories, three annotators). Human ratings are stricter on the absolute scale but strongly agree with GPT-4.1 on the relative method comparison.

(a) Absolute score calibration			
Subset	**Human**	**GPT-4.1**	**Human − LLM**
All samples (n=50)	3.11	4.36	−1.25
MangaFlow-Gemini (n=25)	3.73	4.88	−1.15
Direct-Gemini (n=25)	2.49	3.84	−1.35
(b) Agreement			
Human–LLM rank correlation (Spearman)			0.562
Inter-annotator within-one-point agreement			84.0%
(c) Paired preference over 25 stories			
Averaged human judgement (MangaFlow / Direct / tie)			24 / 1 / 0
GPT-4.1 judgement (MangaFlow / Direct / tie)			17 / 0 / 8
Human–LLM no-conflict agreement			24/25 = 96.0%



Figure 8: Visualization of layout self-reflection in MangaFlow. Each row shows one example, where the left page is generated using the original layout and the right page is generated using the self-refined layout. The reflected layouts better balance panel allocation and page-space usage while preserving the original story content.

A.4 Discussion and Limitations

MangaFlow demonstrates that controllable manga generation benefits from explicit intermediate structures, but several limitations remain. First, the final visual quality still depends on the underlying image generation backbone, so failures in character rendering, fine-grained actions, or local visual details may affect the generated panels. Second, although MangaFlow introduces explicit layout control and lettering, speaker-aware bubble placement remains challenging in complex panels, especially when stylized manga faces are difficult to localize. Third, MangaGen-MetaBench is designed as a meta-benchmark built on existing story visualization data, rather than a fully annotated manga dataset with human-designed layouts, bubble positions, and speaker annotations.

These limitations suggest several future directions. More powerful reference-conditioned renderers could further improve character and scene consistency. Manga-specific perception models may help improve face localization, speaker grounding, and bubble placement. Richer manga annotations could also support more comprehensive evaluation of layout design, reading flow, and lettering quality. Despite these limitations, MangaFlow provides an initial step toward a controllable agentic paradigm for manga generation.