DreamingComics: A Story Visualization Pipeline via Subject and Layout Customized Generation using Video Models

Supplementary Material

1. Overview

This document provides additional technical and experimental details for *DreamingComics*, expanding upon the content of the main paper. It is organized as follows:

- **Section 2** provides inference details on our image customization model, Dream-Illustrator, and compares the inference time and memory against other video-based methods [5, 21].
- **Section 3** describes the dataset construction pipeline, filtering criteria, and ethical considerations.
- **Section 4** provides metric definitions and evaluation protocols for ViStoryBench [45].
- **Section 5** reports comprehensive ablation studies analyzing the contributions of each module.
- **Section 6** includes high-resolution qualitative comparisons, layout visualizations, and failure cases.
- **Section 7** outlines the user study setup, questionnaire design, and statistical analysis.
- **Section 8** discusses remaining challenges and future work directions.

2. Dream Illustrator

2.1. Inference Details

During inference, we preprocess the input subject image by removing its background using RMBG 1.4 [3], so that the image's background does not affect the image customization. We resize the image's width and height to be divisible by 16. If the subject image's style is close to photorealistic and depicts a realistic human, we crop the facial region from the subject [38] and use it as the subject image, enabling a wider range of possible poses and clothing while remaining faithful to the facial identity. If the subject image is of an unrealistic style or depicts a deformed character, we use the full subject image with its background removed. When a layout box is smaller than the reference latent resolution, the RoPE index spacing is uniformly subsampled to prevent phase aliasing, preserving smooth positional variation and ensuring stable, artifact-free positional encodings across varying layout sizes.

Attention Masking for Multi Subject. Motivated by Eligen [41], we design an attention masking strategy when multiple subject inputs are given. To prevent information leakage between two distinct subject latents c_i and c_j, we craft an attention mask M that assigns an extremely negative value to cross-attention scores between different latents, i.e.,

$$M(p,q) = \begin{cases} -\infty & \text{if } (p,q) \in (c_i \times c_j) \\ 0 & \text{else} \end{cases}$$

Note that, unlike prior methods [7, 41], we do not assign "hard" regional attention maps according to the layout condition, as we found that they may harm the visual consistency of our pipeline. We illustrate this further in our ablation section (Section 5).

Method	Inference Time (seconds)	Memory Used (GB)
RealGeneral [21]	92.58	25.91
DRA-Ctrl [5]	58.46	46.34
Ours (Size : 256×256)	12.85	43.13
Ours (Size : $1,280 \times 720$)	17.05	43.99

Table 1. Average inference time per image on a single H100 GPU. All inferences were based on the official code.

2.2. Inference Time and Memory Cost

We provide the inference-time comparison for our method with other video-based methods [5, 21] at Table 1. Here, we measure the computation time to generate an image of 1280×720, for a single given subject. Since RealGeneral [21] and DRA-Ctrl [5] require the subject size to be equal to the target size, we resize the subject size to 1280×720 for these methods, while we test for different subject sizes: (256×256) and (1280×720). All computations were performed on a single H100 GPU. As shown in Table 1, our method can significantly outperform other methods in runtime; we outperform DRA-Ctrl by more than 3 times and RealGeneral by more than 5 times. We owe this to the use of FramePack [42], which allows for the generation of a single output image while fully utilizing the video model priors.

3. Dataset Construction and Ethics

3.1. Comics Layout Dataset

We list the different comic datasets that were used for fine-tuning our Qwen-based [36] layout generator in Table 2. After annotating the panel and character bounding boxes, either using the officially annotated boxes or the

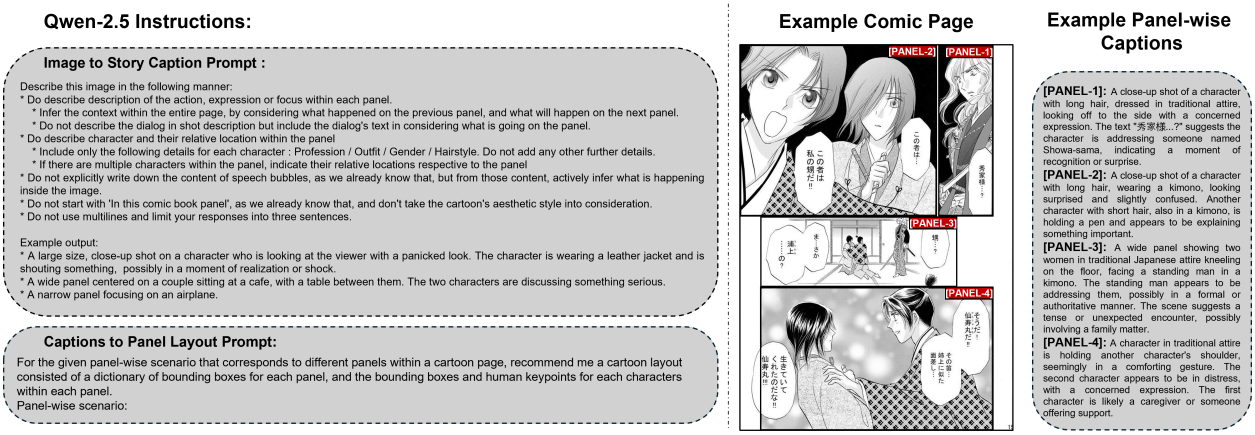


Figure 1. (top-left) Instruction Prompt used for obtaining panel-wise captions for each comic dataset. (bottom-left) Prompt used for fine-tuning Qwen [36] to output the corresponding layout to a set of captions. (right) An example from our processed comic dataset, with a spatial layout for a single comic page (left) and its corresponding caption (right).

Magiv2 panel detector [29], we select and clean a subset of comic pages that constitutes a stable distribution for training. Specifically, we only include pages that consist of more than 2 panels and fewer than 8 panels. For each panel-wise character bounding box, if the relative size of the box to the panel is less than 5% or larger than 90%, we remove that character box from the list. If the size is less than 20% or larger than 75%, we resize the box to an appropriate size within 20% to 75%. We also remove boxes that overlap by more than 25% with another bounding box. This minimizes the risk of erroneous detection results from the panel detector.

Once the panels and bounding boxes are processed, we use Qwen-2.5-VL-7B [36] to process the panel-wise captions. We first use panel-wise images, along with an instruction prompt, as input to generate story captions for each panel. We then pair these captions with the cleaned layouts to form a question-answer pair for supervised fine-tuning. Prior to tuning, we also run a prompt that, for each panel caption, identifies the number of valid characters within the panel and filters out the layouts with mismatching character numbers to ensure consistency. For training and inference, we attach the number of characters (e.g., 2 characters) in front of the input caption to encourage the model to produce layouts with the correct number of character bounding boxes. In total, we process 35,702 layouts and use 35K layout-caption pairs for training and 702 pairs for testing.

3.2. Paired Subjects Dataset

We list the different datasets and preprocessing procedures for fine-tuning FramePack[42] for next frame prediction. The OpenS2V-Nexus [40] dataset consists of 13 subject-text-video triplet sets. Each set includes aesthetic [31] and motion scores [2] for the video, frame-wise facial bounding boxes detected by an image-prompt detection model [11], a multi-modal retrieval model [43], segmentation masks from GroundingDINO [20], and captions generated using Qwen-2.5 [36]. For a given set, we first select videos that are in the top 25% for both aesthetic and motion scores. To further filter videos with at least one visible, consistent human entity, we identify video bounding boxes that occupy more than 1% of the total frame and do not overlap with other boxes. This filtering selects videos with clear, unobscured identities and an appropriate amount of motion so that the human does not appear static throughout the video but does not show dynamic actions that distort identity. The layout of the first frame is then acquired using the provided segmentation mask, and a source frame is selected from the video frames that are at least 1 second apart from the first frame, forming a source-target frame pair. Finally, we measure the TopIQ scores [8] for the target and source frames and only select the pair if both scores are above 0.3.

For the Anime-Shooter [26] dataset, we were able to download 2,915 videos from YouTube that were listed in the dataset metadata. Using the character-level segmentation masks included in the metadata, which were extracted using Sa2VA [39], we extract the source reference images and target latent masks for randomly selected frame pairs in a manner similar to OpenS2V-Nexus. We then measure the TopIQ scores to filter the frame pairs. For the Subject200K dataset [33], we use the text metadata to get the subject information, followed by LISA [19] to predict the mask of the subject with the name in the target image, which is converted into a single-subject layout. In total, we processed 56,691 pairs of single-subject and layout paired data, and we used 55K pairs for training. We processed

Method	Selected Pages	Annotated Pages	Annotated Panels	Annotated Characters
COMICS [17]	22,864	197,466	1,075,393	2,981,509
Manga109 [14]	11,118	16,051	83,748	131,759
PopManga [28]	1,720	1,925	9,578	18,778

Table 2. List of datasets used for training our layout generator.

21,027 pairs of multi-subject and layout paired data, and we used 20K pairs for training.

3.3. Copyright Details and Ethical Considerations

Both OpenS2V-Nexus [40] and Anime-Shooter [26] publicly released their datasets under terms that limit their use to academic research. OpenS2V-Nexus was released under the CC-BY-4.0 License, hosting all of its data exclusively for registered users. Anime-Shooter was released under the CC-BY-NC 4.0 License, providing the annotated scripts, metadata, and YouTube links instead of the actual videos. Subjects200K [33] is a synthetic dataset built using FLUX [1] and was released under the Apache License.

The COMICS [17] dataset consists of American comics published between 1938 and 1954, gathered from the Digital Comics Museum, which hosts user-uploaded scans of copyright-expired, public domain comics. The Manga109 [14] consists of Japanese comic books (Manga) for which permission was granted by the authors of each of the works under the condition that it will be used solely for academic purposes. PopManga [28] consists of English-translated Japanese Manga, collected from various data sources [4, 23], and is not assumed to be used for commercial purposes.

We have strictly used these resources in accordance with their intended usage and have not employed them for any commercial applications.

4. Quantitative Evaluation Details

4.1. ViStoryBench Details

ViStoryBench is a relatively novel benchmark for story visualization, a field that has lacked a comprehensive, story-centric evaluation benchmark. To account for various story types and artistic styles, ViStoryBench curated 80 story segments with 344 characters to balance narrative structures and visual elements. Each story conveys diverse characteristics, including different plots (such as comedy and horror) and visual aesthetics (such as anime and 3D renderings).

4.2. ViStoryBench Metrics Definition

In order to establish a comprehensive story visualization benchmark, ViStoryBench [45] has also quantified several attributes important for story visualization and solidified those attributes as evaluation metrics. Here, we try to follow the implementations provided in ViStoryBench's official repository as closely as possible.

(a) Output (without loss) (b) Attention (without loss) (c) Output (with loss) (d) Attention (with loss)

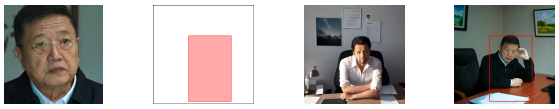
Figure 2. Attention maps between our model trained without and with masked condition loss, marked with input layout. Our new loss helps to position the attention around the target layout, which is evident in the attention map in (d) (Transformer layer = 2), and the resulting image in (c), naturally inducing the model during training to position the character.

Character Identification Similarity (CIDS). The CIDS score assesses the resemblance between the generated characters and the reference characters (cross), as well as the consistency within the generated characters themselves (self). We first use Grounding DINO [22], an open-set object detector, to crop specific characters from the reference image and the generated images based on the character description prompts. If GroundingDINO fails to return the detected region for a character, the score becomes zero. We then extract the character features from the cropped image; depending on the story type, if the character is realistic, we use an ensemble of ArcFace [12], AdaFace [18], and FaceNet [6] for feature extraction; otherwise, we use CLIP [27]. After feature extraction, we compute the cosine similarity between the 512-dimensional feature vectors to obtain the character similarity score. Finally, we average the results by story and return the final averaged results for cross-similarity and self-similarity.

Style Similarity (CSD). The CSD score assesses the stylistic consistency between the generated images and the reference images (cross) and the consistency within the generated images themselves (self). We use the CSD-CLIP [32] feature analysis, which employs a CLIP [27] vision encoder pre-trained on a large-scale style dataset and returns the style features disentangled from the content features. We then compute the cosine similarity between the feature vectors for both cross-similarity and self-similarity.

Onstage Character Count Matching (OCCM). The OCCM score checks whether the generative pipeline generates the correct set of intended onstage characters, without any superfluous characters (unexpected additions) or omissive characters (failure to render specific roles). Based on the detected characters from the CIDS stage, the OCCM score is calculated as:

$$\text{OCCM} = 100 \times \exp\left(-\frac{|D-E|}{\epsilon+E}\right) \tag{1}$$



(a) Character Image (b) Layout (c) Output (ROICtrl [16]) (d) Output (Ours)

Figure 3. Example generation for ViStoryBench, where given a target character image, layout and text, we generate a corresponding story frame for different methods.

where D denotes the detected number of characters, E represents the expected character count from storyboard specifications, and $\epsilon = e^{-6}$ is the smoothing factor to prevent division by zero. Because the code for OCCM was not explicitly defined within the code repository while being written in the paper, we implemented our scoring code that follows the official description.

Layout Precision. To correctly measure our method's subject-wise layout fidelity to other layout-controlled methods, we introduce a *Layout Precision* metric that quantifies the spatial containment of predicted bounding boxes in relation to the input bounding boxes. This metric focuses on whether predictions remain within valid ground-truth regions, rewarding predictions that lie entirely inside the layout boxes while penalizing only those that extend beyond them, matching our method's intuition of giving 'soft' bounding boxes as inputs rather than 'hard' boundaries. Based on the detected characters, we select character boxes that have a confidence score of over 0.5. For each predicted box, we compute the ratio between the intersection area with its best-matching ground-truth box and the total area of the predicted box, emphasizing the spatial precision of the character-wise box. We then calculate this score for all available characters and average it.

Copy Paste Detection. The Copy-Paste detection score identifies whether the generative pipeline has directly "copy-pasted" the character reference image for generation, which is undesired for story visualization. Specifically, for methods limited to single-image input, and for stories that provide multiple reference images for a single character, the metric computes the disparity in similarity between the output characters and both the input image and an alternative reference image. A higher value will indicate a greater propensity for the model to replicate characters directly from reference images. Since our method and other image customization methods like UNO [35] and DreamO [24] support multiple image inputs for a single subject, we do not explicitly show this in our paper.

Image Quality Metrics. We calculate the aesthetic quality score [13] for all generated results and average them. We also calculate the Inception score [30] for all generated results.

4.3. Adapting Methods to ViStoryBench

We list the specifics of adapting each method to the ViStoryBench evaluation. When possible, we used the inference codes for each method provided in the official ViStoryBench repository. We also list the following specifications for each method. Methods not listed below followed the same procedure as detailed in the official repository.

Copy-Paste Baseline. ViStoryBench provided the scores for the Copy-Paste Baseline, which involves directly copying and pasting characters into images. We include these scores in our quantitative comparison to assess how closely other methods represent the aesthetics of the overall characters and stories.

Story Adapter. We report the results for the setting image-ref, scale 5, which was the main result used for reporting in ViStoryBench. [45]

DiffSensei. We followed the original implementation and weights of DiffSensei [34] and used the same layout conditions generated from our layout generator. Due to the inference memory issue, we used the inference demo that does not utilize the MLLM model [15], which is claimed to 'largely unaffected the overall quality'.

Eligen. Since Eligen [41] does not support input reference images, we used the prompt of each shot of ViStoryBench dataset and the same layout conditions to generate images.

DreamO. We follow ViStoryBench's official implementation of inferencing UNO [35] and adapt its structure to DreamO [24].

RealGeneral. We followed the subject-driven generation pipeline provided by the original implementation, with an aspect ratio fixed to 16:9; i.e., 1344×768. Since RealGeneral only accepts a single reference image with the same size as the target, we select the first reference image for each shot and resize it to 1344×768 before passing it to RealGeneral.

DRA-Ctrl. We followed the subject-driven generation pipeline provided by the original implementation, following the size limit of 512×512 imposed by the official code for each ViStoryBench story.

5. Ablation Studies

5.1. Masked Condition Loss Strength

To evaluate the influence of the masked condition loss, we vary $\lambda_{\text{MASK}} \in \{0.0, 0.05, 0.1, 0.25\}$ under the same training setup used in our main ablation study and report CIDS and CSD scores on ViStoryBench. As shown in Table 3, removing the loss entirely ($\lambda_{\text{MASK}} = 0$) leads to degradation in both identity and style consistency, reflecting uncontrolled attention spillover across subjects. Increasing λ_{MASK} beyond a small value does not improve performance: both 0.10 and 0.25 reduce CIDS and

Name	CIDS (Cross/Self)	CSD (Cross/Self)
($\lambda_{\text{MASK}} = 0.0$)	56.6 / 61.2	50.5 / 56.3
($\lambda_{\text{MASK}} = 0.1$)	53.8 / 56.7	48.9 / 56.1
($\lambda_{\text{MASK}} = 0.25$)	54.5 / 59.0	49.4 / 56.3
Ours ($\lambda_{\text{MASK}} = 0.05$)	**58.4 / 63.0**	**52.9 / 62.4**

Table 3. Ablation study on the impact of choosing the right timestamp for the target image.

Name	CIDS (Cross/Self)	CSD (Cross/Self)
with Regional Attention	49.2 / 55.8	44.7 / 55.0
Ours	**58.4 / 63.0**	**52.9 / 62.4**

Table 4. Ablation study on the impact of choosing the right timestamp for the target image.

CSD, suggesting that stronger penalties hinder global context integration. Our chosen setting, ($\lambda_{\text{MASK}} = 0.05$), achieves the best overall results, indicating that a light spatial constraint is sufficient to stabilize multi-subject disentanglement without sacrificing stylistic coherence.

5.2. Regional Attention Mask

For better comparison, we also experiment with a regional attention mask adapted from previous literature [7, 41] to determine whether its presence positively affects quality. As shown in Table 4, results with the regional attention mask lead to worse overall identity and style preservation compared to our constraint formulation. Overall, combining RegionalRoPE with our soft training loss yields the best results, validating our design choice of using learnable layout constraints rather than rigid attention control, leading to more accurate and flexible subject placement.

6. Additional Qualitative Results

6.1. Image Customization Results

From Fig. 4 to 7, we present the results from our image customization pipeline, Dream-Illustrator, alongside the results from other competing methods [21, 24, 34, 35, 41, 44] based on the same character images and layouts. Previous layout-guided methods [34, 41] can accurately position characters within a given layout, but they fail to support reference identity and artistic style. Other methods also tend to be biased towards photorealistic styles, neglecting the diverse art styles unique to the reference image. On the contrary, our methods can support diverse types of images and story types, such as children's picture book illustrations (Fig. 4, Fig. 7), Minecraft-style animations (Fig. 6), and realistic sewing-doll puppetry (Fig. 5).

From Fig. 8 to 11, we present comic-level story visualization results from our full pipeline, where we first generate layouts from the input panel-wise captions and use them, along with the reference images and captions, to synthesize a full-length comic-style story. The results capture diverse types of art styles, from cel-style animation to photorealistic dramas, and correctly position the given reference images within the layout regions while adhering to the panel-wise caption context. In Fig. 12, we present a story-level comparison between our pipeline and other competing methods. We show that our pipeline can consistently create images for the given characters without omitting or adding any characters not specified by the input instructions.

6.2. Layout Generation Visualization

Fig. 13 to 16 visualizes the results from our layout generator pipeline, which, given a multi-panel caption (T_i), produces structured panel (D_i) and character bounding boxes (BOX_i). Fig. 13 and 15 show the diverse layouts generated from the same caption set, demonstrating the flexibility of our layout generator for story visualization while remaining faithful to the given prompt. Fig 14 and 16 also show the different layouts generated by stories from ViStoryBench [45].

Fig. 17 shows the ability of Dream-Illustrator to control the character position according to the given layout, without losing character identity and artistic style. Fig. 18 demonstrates how robust our method can be to different shapes of layouts, preserving the aspect ratio of the reference image's aesthetics without any distortions.

7. User Study Details

From Fig. 20 to Fig. 23, we present the screenshots for each question type in the user study.

We conducted a user study with 26 participants and a total of 20 questions to evaluate our story visualization pipeline in terms of image customization and layout generation. The first 15 questions compared our image customization method with two other state-of-the-art methods, UNO [35] and DreamO [24], in terms of visual consistency. The last 5 questions compared our layout generation method with GPT-4 [25], in terms of creating plausible layouts.

Specifically, the first 5 questions evaluate how closely the methods preserve the identity of the input image. Each participant assessed the input reference image along with images generated by both baseline methods and our approach. The evaluation question was

- The two images below were made based on the original image shown above. Which one looks **more like the same character** from the original image?

The next 5 questions evaluate how closely the methods preserve the style of the input image. Based on a similar format to character identity, the evaluation question was

- The two images below were made based on the original image shown above. Which one **better keeps the artistic style or feeling** of the original image?

The next 5 questions evaluate how closely the methods maintain visual consistency throughout story visualization. Instead of showing the reference image, a series of 4 images for a single story, generated by both baseline methods and our approach using the same reference image, was displayed to each participant. The evaluation question was

- Each set below shows four images created by the same method. Which set of images **keeps a more consistent art style** across all four pictures?

The last 5 questions evaluate the plausibility of our layout generator method in creating layouts that consist of the correct number of panels and characters, while remaining faithful to the customs of traditional comics. Each participant assessed the input panel-wise captions along with the visualized layouts generated by both GPT-4 [25] and our approach using the same captions. The evaluation question was

- The two images below show different comic page layouts created from the same script. Which layout looks **more natural and accurate** for the story—for example, with the right number of panels and characters, and a clear, well-arranged page?

8. Limitations and Future Work

Although DreamingComics demonstrates strong layout-aware customization, its performance may degrade when the layout input is poorly specified. Despite using positional guidance rather than enforcing strict masking constraints [41], errors such as missing or misaligned bounding boxes—e.g., in multi-subject scenes—can still result in incomplete or incoherent outputs. Faulty configurations, such as overlapping or irregular character regions, may further reduce visual fidelity (see Fig. 19). Although our code applies rule-based corrections (e.g., ignoring bounding boxes below a certain size), unreliable layouts can occasionally require users to provide more carefully crafted inputs, which can be a tedious task and negatively impact the user experience.

Badly structured layouts can also impact the model's prompt adherence. Since our method jointly conditions on text, image, and layout—unlike most prior works that consider only text and image—conflicts between the textual prompt and the layout may lead to outputs that do not reflect the intended narrative. These failure cases are further influenced by the fact that our model is fine-tuned from a pretrained video generator using a relatively compact, video-derived dataset with limited stylistic diversity and layout variability.

Looking forward, we believe that future work can address these issues by scaling both the model and the training data. Enhancing the layout generator with reliability checks or feedback-guided refinement could mitigate downstream spatial artifacts. Also, using larger and more diverse datasets annotated with layout and subject signals, paired with advanced video diffusion backbones, could unlock greater stylistic coverage and layout generalization. With the advent of advanced generative models that can precisely render text [9, 10, 37], we can also add layouts for text boxes, onomatopoeia, and other visual effects that can positively benefit comic creation.

References

[1] Black Forest Labs. FLUX: High-Fidelity Image Generation Model. https://github.com/black-forest-labs/flux?tab=readme-ov-file, 2025. Accessed: 2025-08-01. 3

[2] Gary Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000. 2

[3] BriaAI. Rmbg-1.4. https://huggingface.co/briaai/RMBG-1.4, 2025. Accessed: 2025-09-09. 1

[4] Mangaplus by shueisha. Mangaplus. https://mangaplus.shueisha.co.jp/. 3

[5] Hengyuan Cao, Yutong Feng, Biao Gong, Yijing Tian, Yunhong Lu, Chuang Liu, and Bin Wang. Dimension-Reduction Attack! Video Generative Models are Experts on Controllable Image Synthesis. *ArXiv*, abs/2505.23325, 2025. 1

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74, 2017. 3

[7] Anthony Chen, Jianjin Xu, Wenzhao Zheng, Gaole Dai, Yida Wang, Renrui Zhang, Haofan Wang, and Shanghang Zhang. Training-free regional prompting for diffusion transformers. *arXiv preprint arXiv:2411.02395*, 2024. 1, 5

[8] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024. 2

[9] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: diffusion models as text painters. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2023. Curran Associates Inc. 6

[10] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing thenbsp;power ofnbsp;language models fornbsp;text

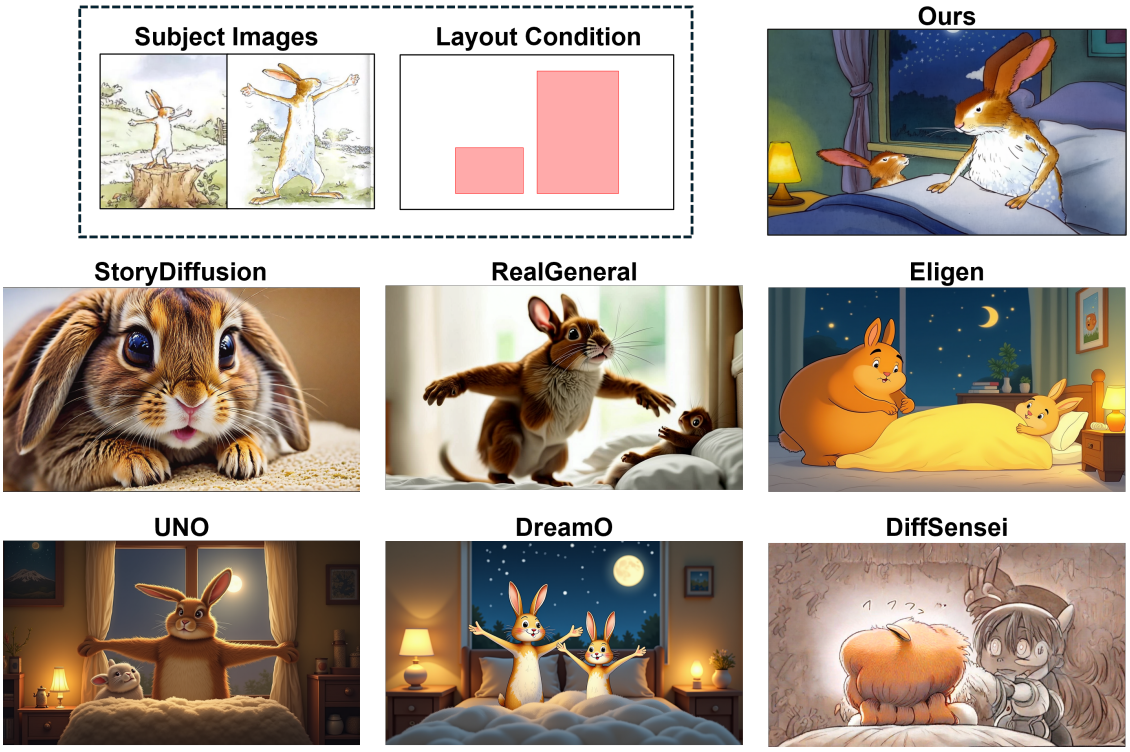


Figure 4. Image-level comparison of our model, which is capable of preserving the character identity and style while controlling locations by layout. Other methods tend to misinterpret the reference character's identity and style.

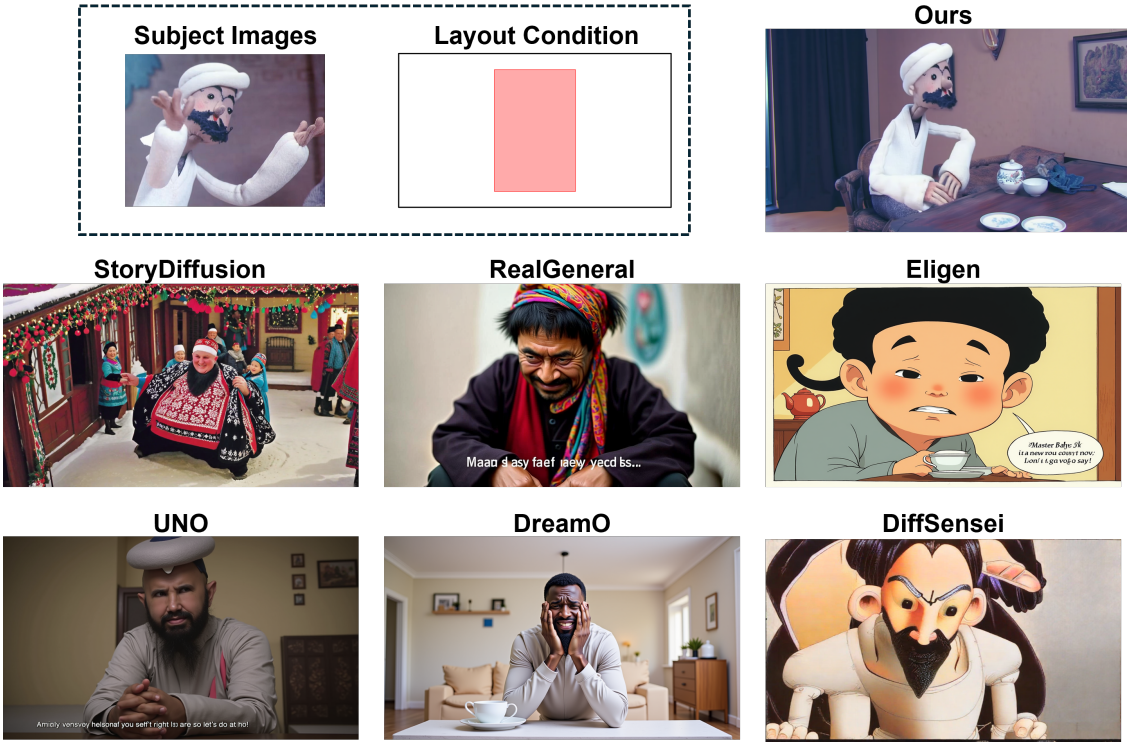


Figure 5. Image-level comparison of our model, which is capable of preserving the character identity and style while controlling locations by layout. Other methods tend to misinterpret the reference character's identity and style.

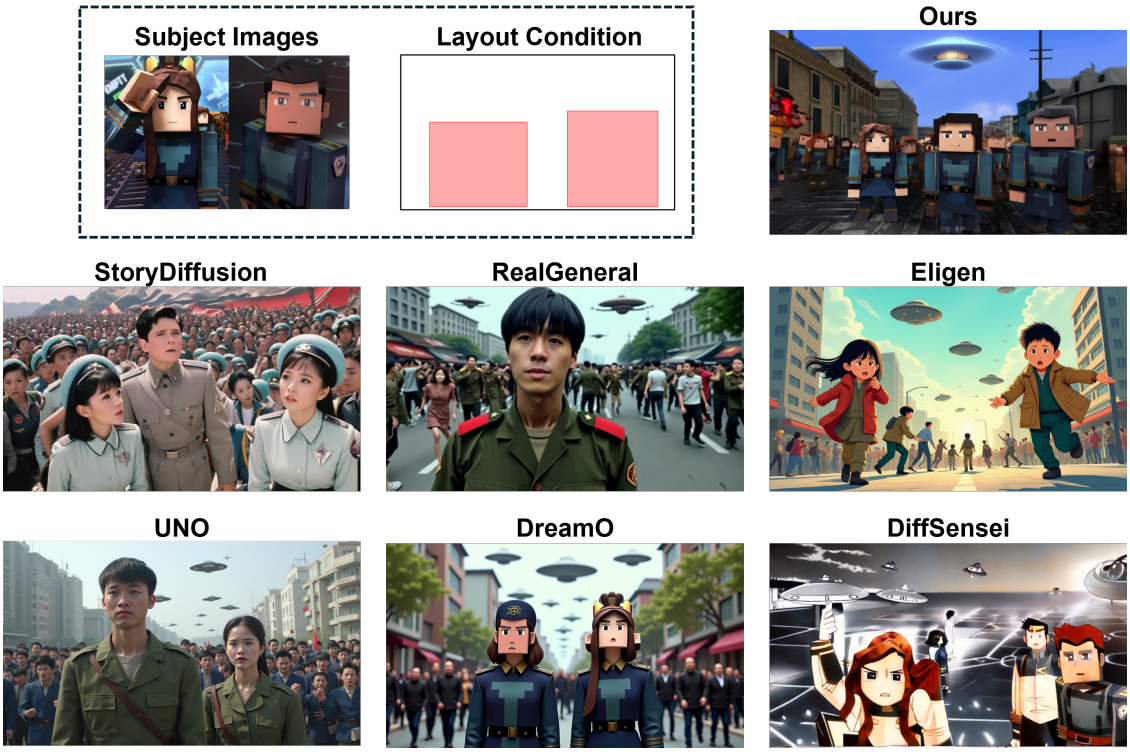


Figure 6. Image-level comparison of our model, which is capable of preserving the character identity and style while controlling locations by layout. Other methods tend to misinterpret the reference character's identity and style.

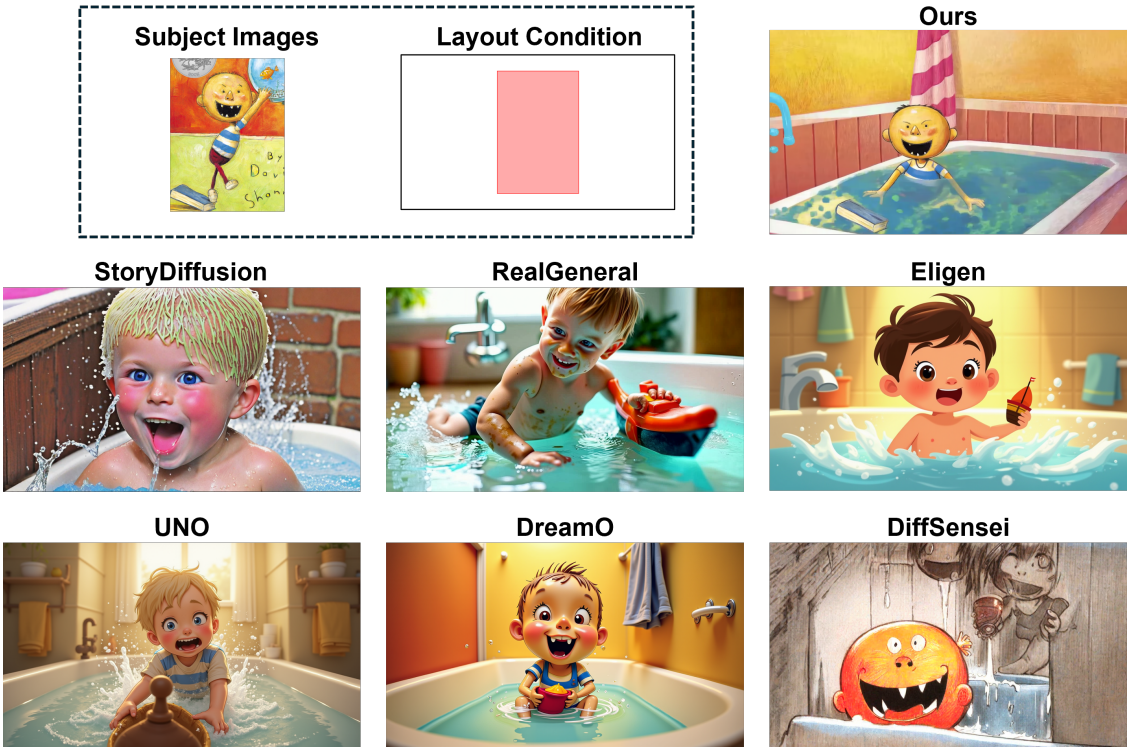


Figure 7. Image-level comparison of our model, which is capable of preserving the character identity and style while controlling locations by layout. Other methods tend to misinterpret the reference character's identity and style.

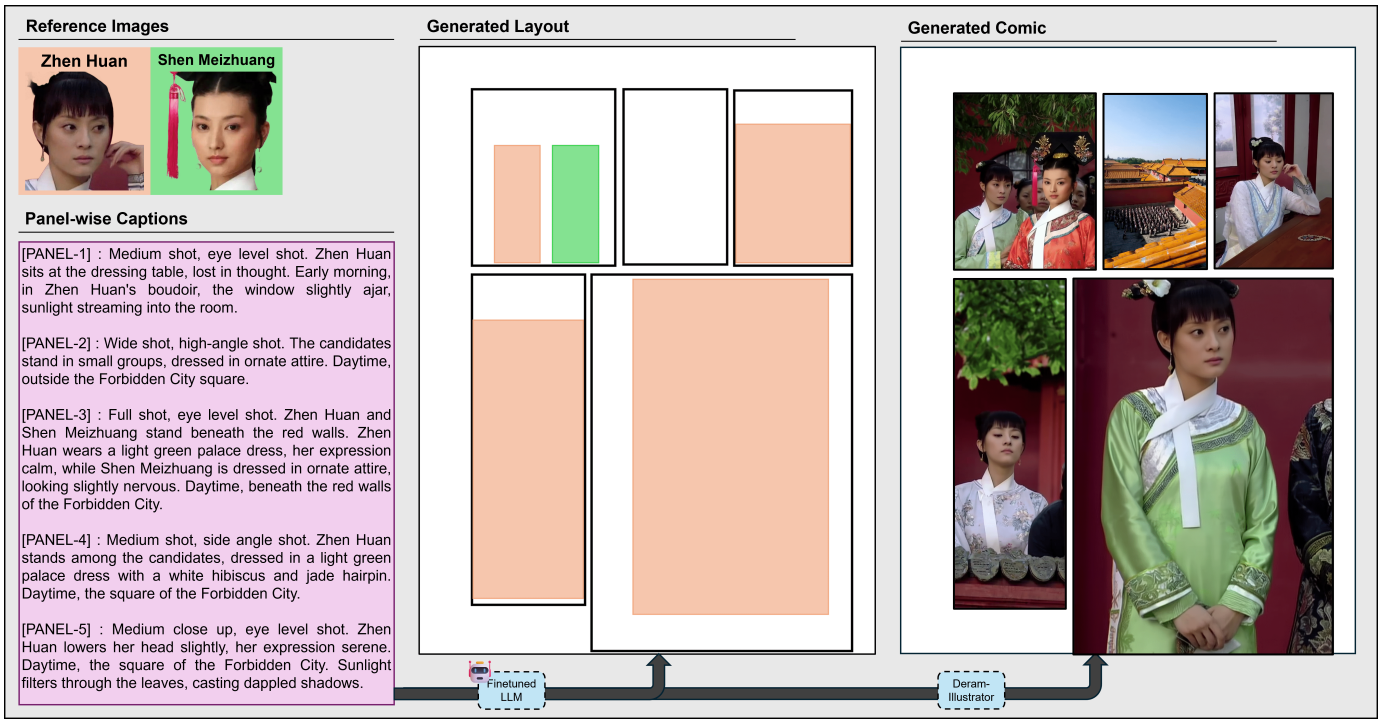


Figure 8. Example story visualization for a 5-panel comic, featuring 2 characters depicted in a realistic style.

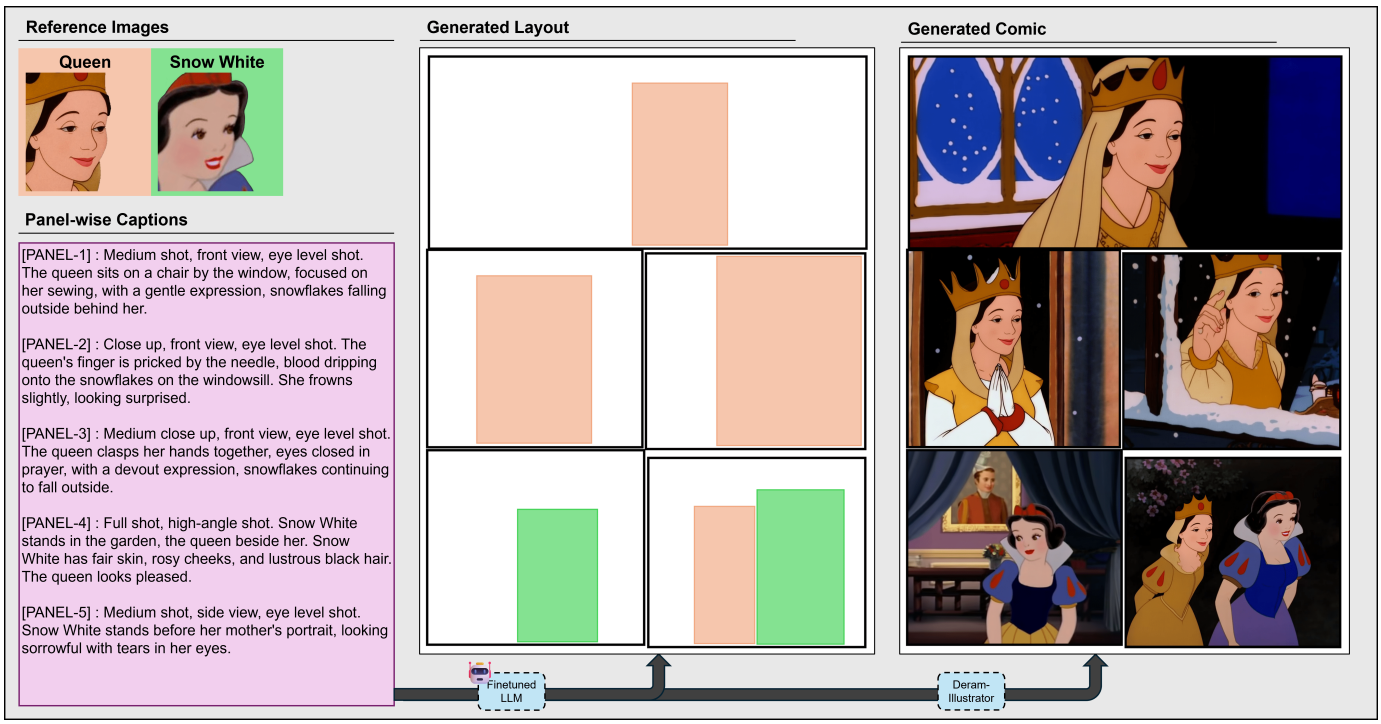


Figure 9. Example story visualization for a 5-panel comic, featuring 3 characters depicted in a traditional animation style.

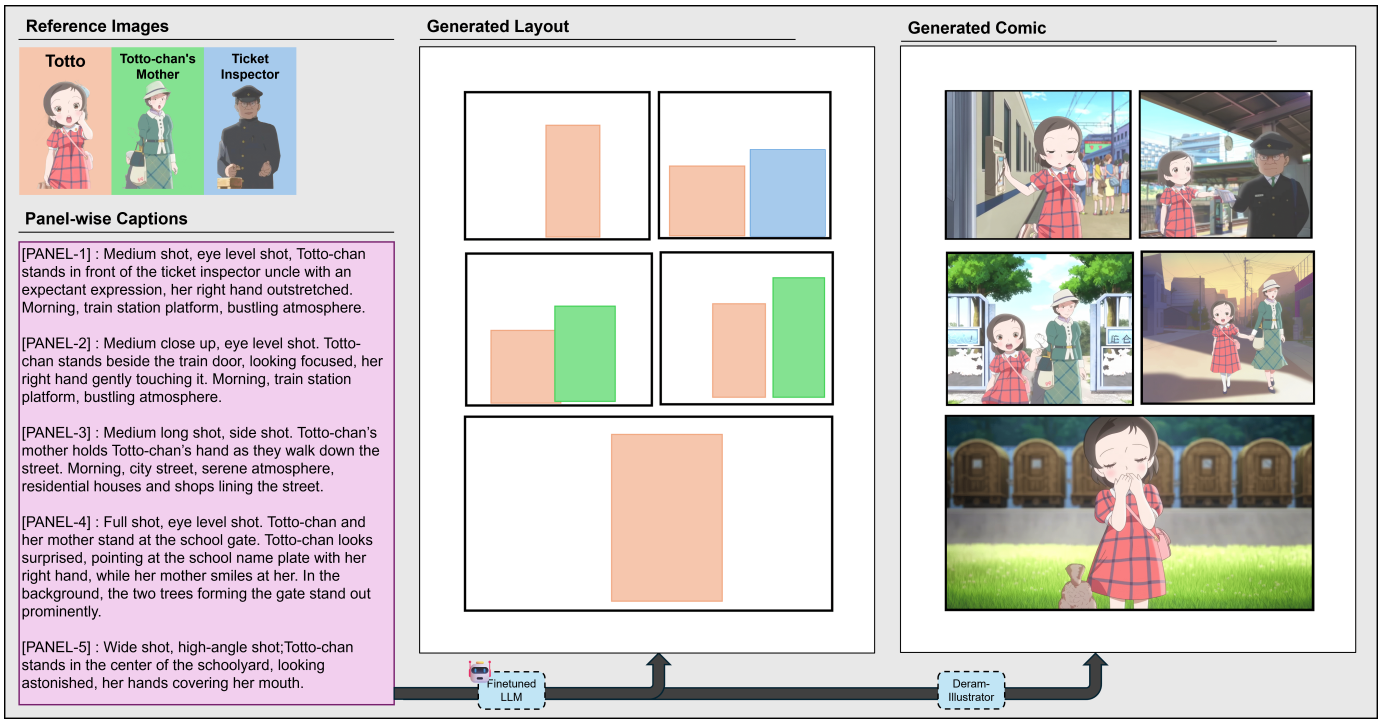


Figure 10. Example story visualization for a 5-panel comic, featuring 3 characters depicted in a Japanese cel animation style.

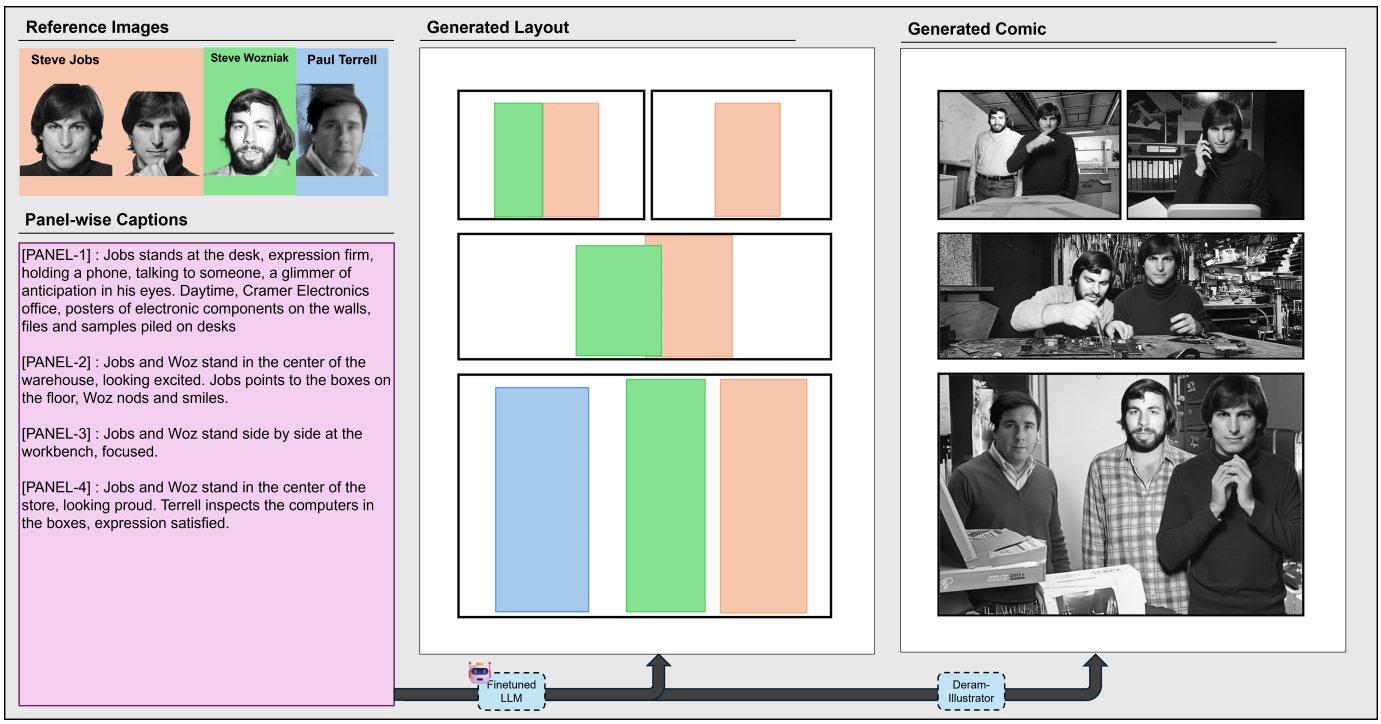


Figure 11. Example story visualization for a 4-panel comic, featuring 3 characters, one with multiple reference images, depicted in a black-and-white photo style.

Figure 12. Qualitative comparison on ViStoryBench. Reference images of each shot's on-stage characters are shown in Copy-Paste baseline results. Compared to other methods, our method demonstrates visually consistent results in both character and style, while adhering to the given captions without omitting or creating superfluous characters.

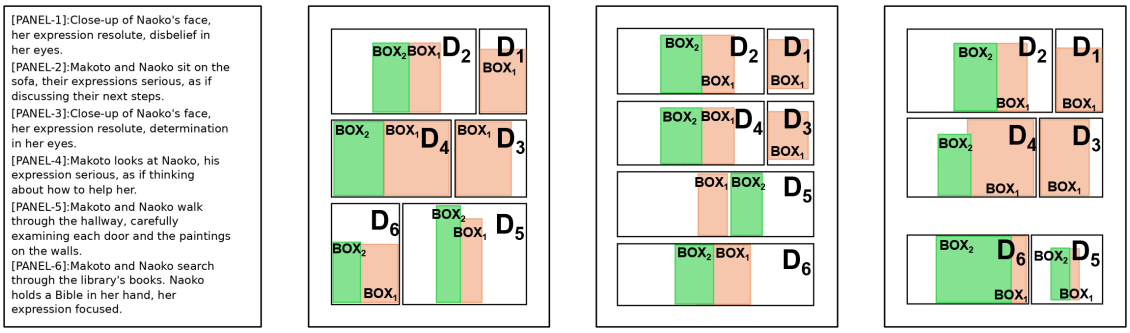


Figure 13. From a multi-panel caption (left), the right three columns show diverse panel ($D_{1:n}$) and character ($\mathrm{BOX}_{1:n}$) arrangements from our layout generator, illustrating its ability to generate multiple coherent layouts for story visualization.

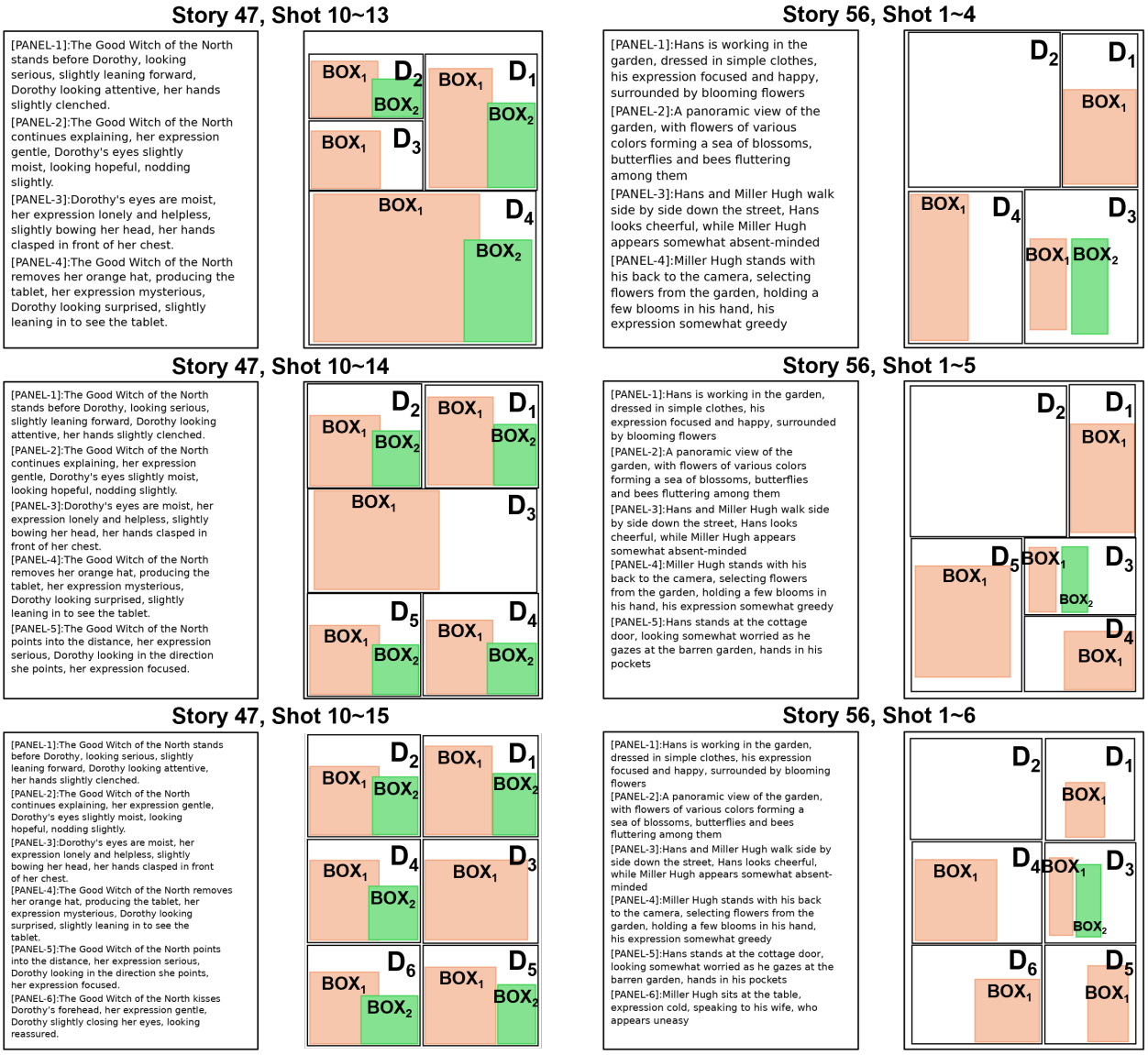


Figure 14. Additional panel / character layouts generated from ViStoryBench [45] stories. Given the same story but with different numbers of preceding panels, our layout generator maintains stable character positions while adapting flexibly to new contexts, indicating strong consistency along with controlled diversity.

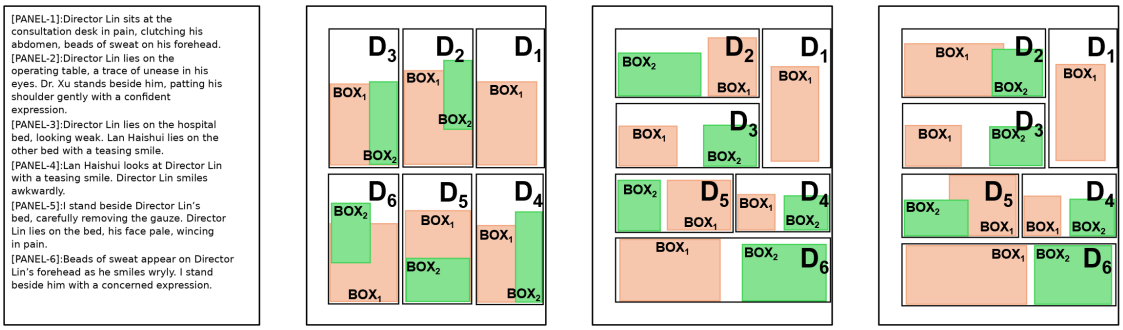


Figure 15. From a multi-panel caption (left), the right three columns show diverse panel ($D_{1:n}$) and character ($\mathrm{BOX}_{1:n}$) arrangements from our layout generator, illustrating its ability to generate multiple coherent layouts for story visualization.

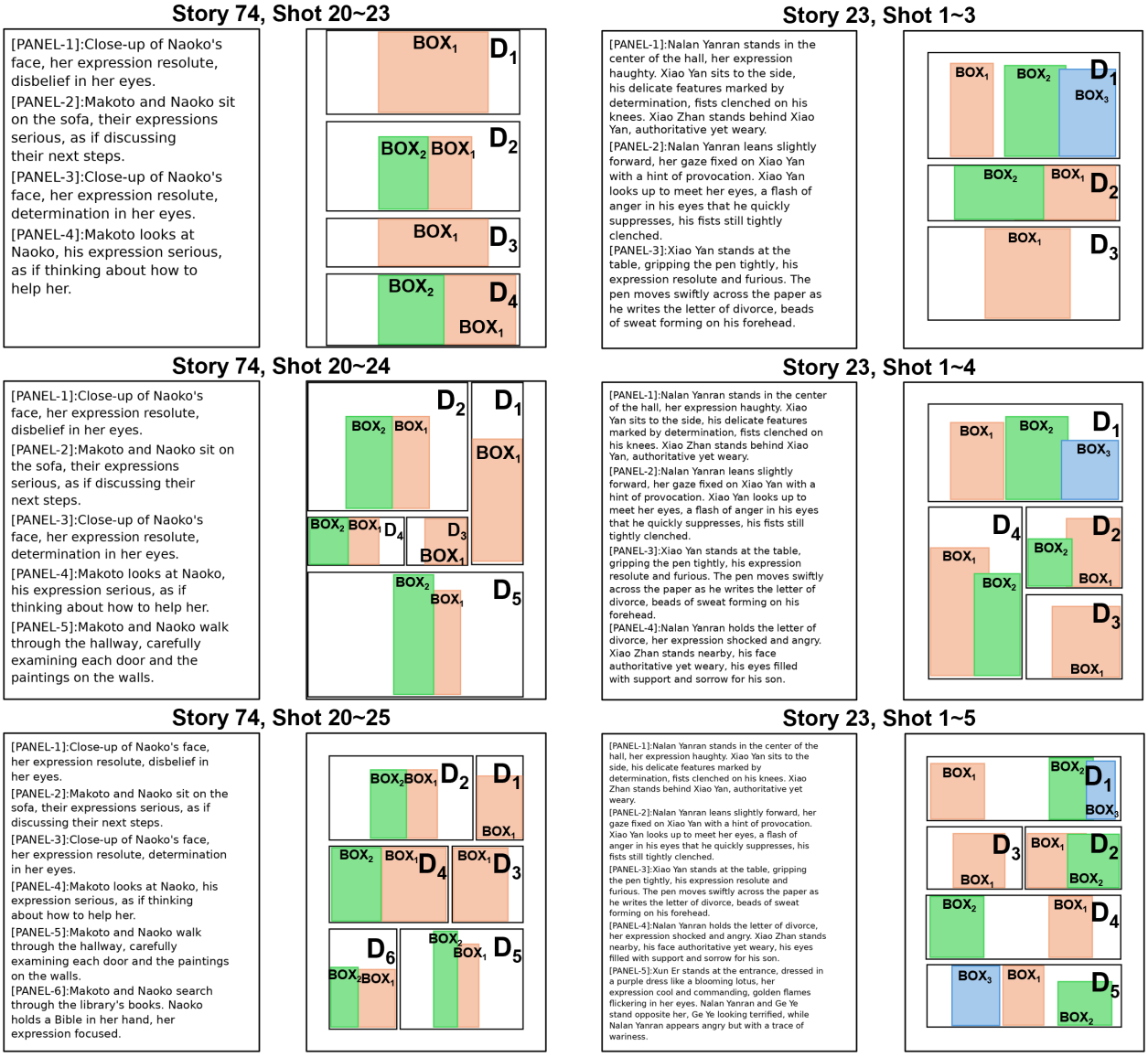


Figure 16. Additional panel / character layouts generated from ViStoryBench [45] stories. Given the same story but with different numbers of preceding panels, our layout generator maintains stable character positions while adapting flexibly to new context, indicating strong consistency along with controlled diversity.

Figure 17. Qualitative results on Dream-Illustrator's ability to control the output by the given layout condition.

Figure 18. Qualitative results on Dream-Illustrator's ability to be robust to layouts that ill-defines the ratio of the actual subject.

Figure 19. **Limitations** Our model fails to support such narrow, impractical layout conditions that are not within the trained distribution, which results in instances where the visual consistency is broken.

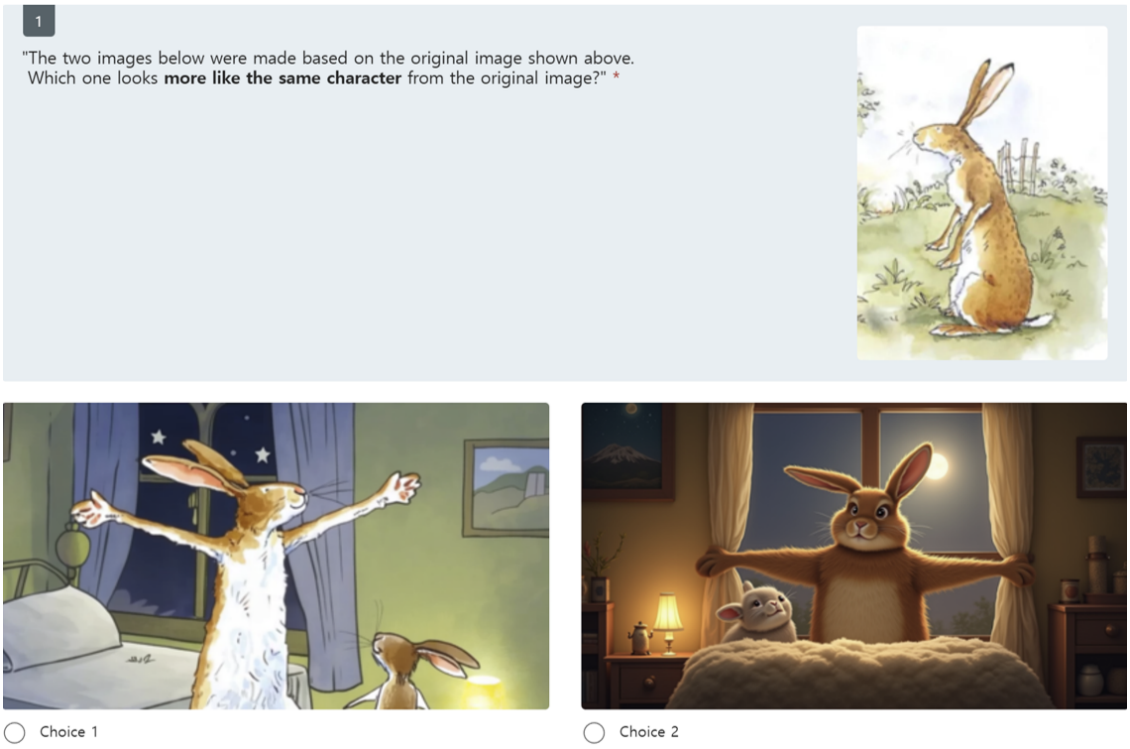


Figure 20. User study interface and question, regarding character identity.

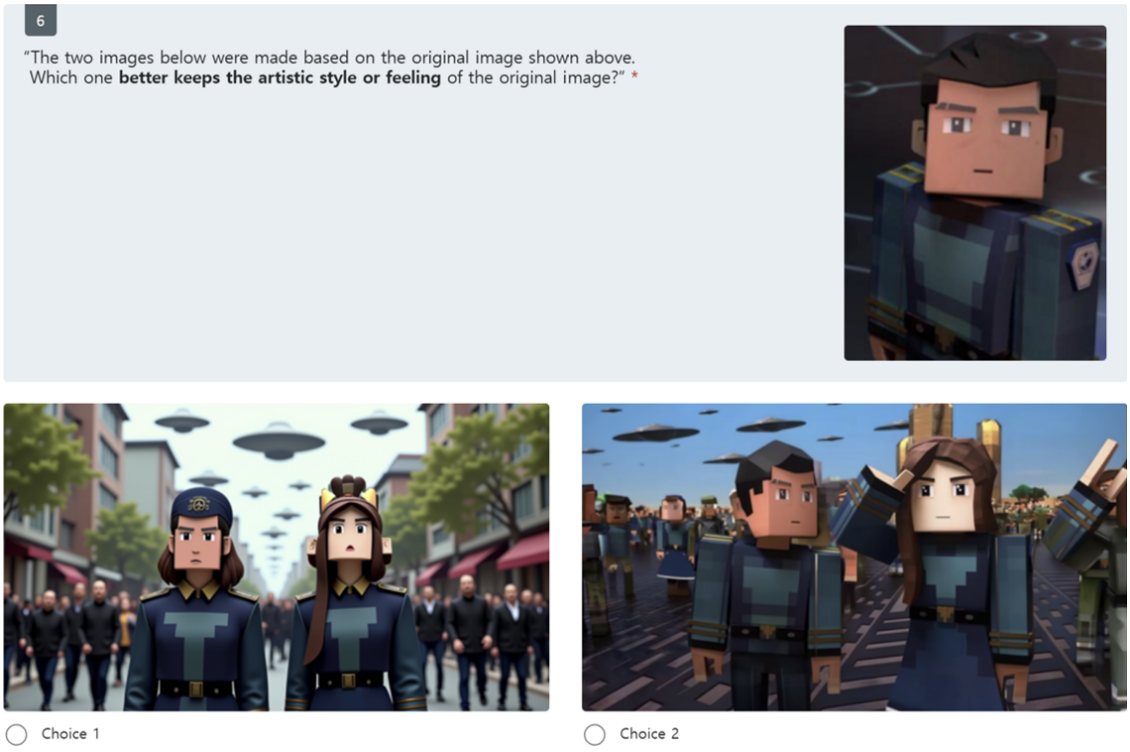


Figure 21. User study interface and question, regarding character style.

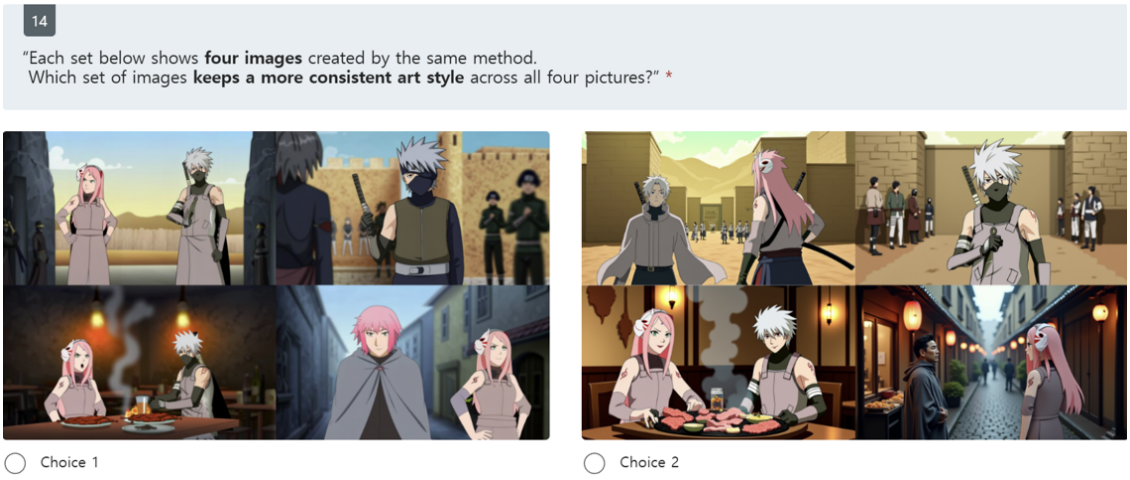


Figure 22. User study interface and question, regarding story consistency.

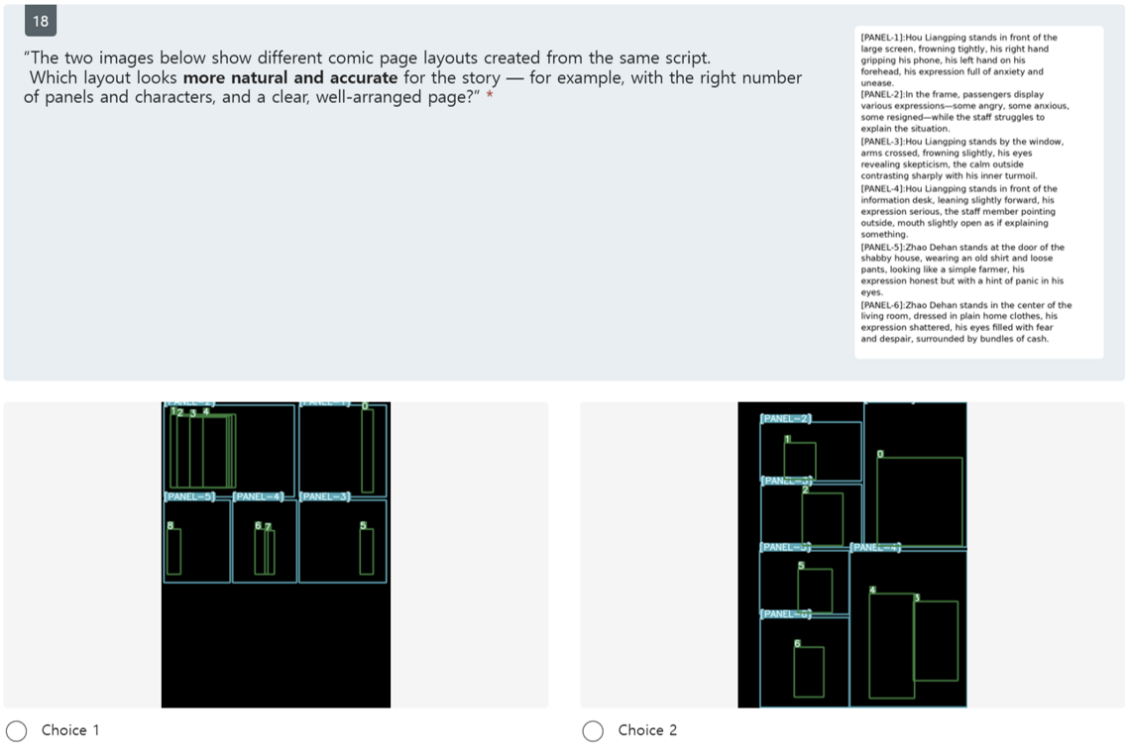


Figure 23. User study interface and question, regarding layout plausibility.

rendering. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part V*, page 386–402, Berlin, Heidelberg, 2024. Springer-Verlag. 6

[11] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16901–16911, 2024. 2

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 3

[13] discus0434. Aesthetic predictor v2.5. https://github.com/discus0434/aesthetic-predictor-v2-5, 2024. Accessed: 2025-08-01. 4

[14] Azuma Fujimoto, Toru Ogawa, Kazuyoshi Yamamoto, Yusuke Matsui, T. Yamasaki, and Kiyoharu Aizawa. Manga109 dataset and creation of metadata. *Proceed-*

ings of the 1st International Workshop on coMics ANalysis, Processing and Understanding, 2016. 3

[15] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *ArXiv*, abs/2404.14396, 2024. 4

[16] Yuchao Gu, Yipin Zhou, Yunfan Ye, Yixin Nie, Licheng Yu, Pingchuan Ma, Kevin Qinghong Lin, and Mike Zheng Shou. Roictrl: Boosting instance control for visual generation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23658–23667, 2024. 4

[17] Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Lee Boyd-Graber, Hal Daumé, and Larry S. Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6478–6487, 2016. 3

[18] Minchul Kim, Anil K. Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18729–18738, 2022. 3

[19] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2

[20] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Trans. Mach. Learn. Res.*, 2024, 2023. 2

[21] Yijing Lin, Mengqi Huang, Shuhan Zhuang, and Zhendong Mao. Realgeneral: Unifying visual generation via temporal in-context learning with video models, 2025. 1, 5

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino withnbsp;grounded pre-training fornbsp;open-set object detection. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLVII*, page 38–55, 2024. 3

[23] Mangadex. Mangadex. https://mangadex.org/. 3

[24] Chong Mou, Yanze Wu, Wenxu Wu, Zinan Guo, Pengze Zhang, Yufeng Cheng, Yiming Luo, Fei Ding, Shiwen Zhang, Xinghui Li, Mengtian Li, Songtao Zhao, Jian Zhang, Qian He, and Xinglong Wu. Dreamo: A unified framework for image customization. *ArXiV*, abs/2504.16915, 2025. 4, 5

[25] OpenAI. Gpt-4 technical report. 2023. 5, 6

[26] Lu Qiu, Yizhuo Li, Yuying Ge, Yixiao Ge, Ying Shan, and Xihui Liu. Animeshooter: A multi-shot animation dataset for reference-guided video generation, 2025. 2, 3

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 3

[28] Ragav Sachdeva and Andrew Zisserman. The manga whisperer: Automatically generating transcriptions for comics. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12967–12976, 2024. 3

[29] Ragav Sachdeva, Gyungin Shin, and Andrew Zisserman. Tails tell tales: Chapter-wide manga transcriptions with character names. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 2053–2069, 2024. 2

[30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc. 4

[31] Christoph Schuhmann. improved-aesthetic-predictor. GitHub repository, 2024. [Online; accessed 16-Nov-2025]. 2

[32] Gowthami Somepalli, Anubhav Gupta, Kamal K. Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. *ArXiv*, abs/2404.01292, 2024. 3

[33] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. 2025. 2, 3

[34] Jianzong Wu, Chao Tang, Jingbo Wang, Yanhong Zeng, Xiangtai Li, and Yunhai Tong. Diffsensei: Bridging multi-modal llms and diffusion models for customized manga generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28684–28693, 2025. 4, 5

[35] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 4, 5

[36] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. 1, 2

[37] Yukang Yang, Dongnan Gui, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, and Kai Chen. Glyphcon-

trol: Glyph conditional control for visual text generation. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[38] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023. 1

[39] Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv pre-print*, 2025. 2

[40] Shenghai Yuan, Xianyi He, Yufan Deng, Yang Ye, Jinfa Huang, Bin Lin, Jiebo Luo, and Li Yuan. Opens2v-nexus: A detailed benchmark and million-scale dataset for subject-to-video generation. *ArXiv*, abs/2505.20292, 2025. 2, 3

[41] Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, and Yu Zhang. Eligen: Entity-level controlled image generation with regional attention. *arXiv preprint arXiv:2501.01097*, 2025. 1, 4, 5, 6

[42] Lvmin Zhang and Maneesh Agrawala. Packing input frame contexts in next-frame prediction models for video generation. *Arxiv*, 2025. 1, 2

[43] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms. *ArXiv*, abs/2412.16855, 2024. 2

[44] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *NeurIPS 2024*, 2024. 5

[45] Cailin Zhuang, Ailin Huang, Wei Cheng, Jingwei Wu, Yaoqi Hu, Jiaqi Liao, Zhewei Huang, Hongyuan Wang, Xinyao Liao, Weiwei Cai, Hengyuan Xu, Xuanyang Zhang, Xianfang Zeng, Gang Yu, and Xiaoyan Zhang. Vistorybench: Comprehensive benchmark suite for story visualization. *ArXiV*, abs/2505.24862, 2025. 1, 3, 4, 5, 12, 13