# Plot'n Polish: Zero-Shot Story Visualization and Disentangled Editing with Text-to-Image Diffusion Models

**Kiymet Akdemir**[1], **Jing Shi**[2], **Kushal Kafle**[2], **Brian L. Price**[2], **Pinar Yanardag**[1]

[1]Virginia Tech, Blacksburg, VA, USA
[2]Adobe Research, San Jose, CA, USA

## Abstract

Text-to-image diffusion models have demonstrated significant capabilities to generate diverse and detailed visuals in various domains, and story visualization is emerging as a particularly promising application. However, as their use in real-world creative domains increases, the need for providing enhanced control, refinement, and the ability to modify images post-generation in a consistent manner becomes an important challenge. Existing methods often lack the flexibility to apply fine or coarse edits while maintaining visual and narrative consistency across multiple frames, preventing creators from seamlessly crafting and refining their visual stories. To address these challenges, we introduce Plot'n Polish, a zero-shot framework that enables consistent story generation and provides fine-grained control over story visualizations at various levels of detail.

**Project page** — https://plotnpolish.github.io

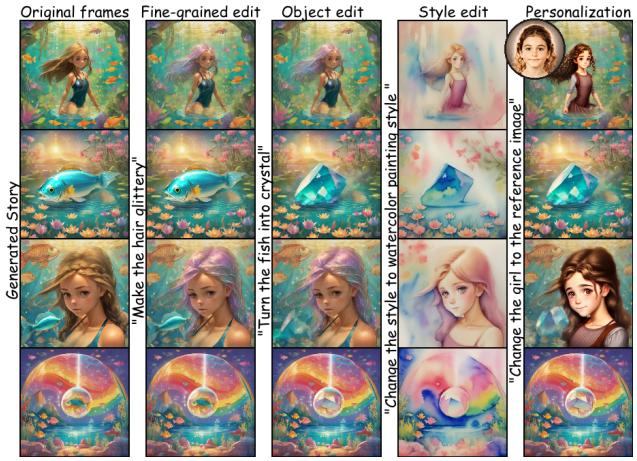


Figure 1: We introduce `Plot'n Polish`, a training-free approach for creating and refining story visualizations. Our framework enables users to adjust story elements through fine or coarse-grained edits. Users can alter elements like hairstyles or clothing, transform objects or styles, and customize characters iteratively directed through text prompts.

## Introduction

Text-to-image diffusion models have emerged as powerful tools for generating high-quality, detailed, and diverse images across various domains (Rombach et al. 2022; Ho, Jain, and Abbeel 2020; Esser et al. 2024). Story visualization is a particularly exciting application, aiming to generate image sequences that form a coherent narrative guided by sequential text prompts. However, as these models are increasingly used in creative processes, there's a significant challenge in providing not only coherent story sequences but also enhanced control and refinement. Storytelling often requires the ability to create and modify the narrative, a flexibility current methods fail to address. Creators might want to alter the storyline by introducing new plot elements or characters during generation. Additionally, after story visuals are generated, creators may wish to make fine-grained adjustments—such as adding accessories to a character or changing attire colors or more substantial changes like replacing a character or altering the visual style of the story.

Current approaches to story visualization face several key limitations (see Table 1), and lack flexibility for post-generation edits, forcing users to regenerate entire sequences to make changes. Many methods demand extensive, computationally expensive training on large datasets (Tao et al. 2024), making them less accessible to a broad range of users. Additionally, several existing techniques generate each story frame independently, leading to visual inconsistencies, particularly when characters, objects, or settings need to evolve fluidly across multiple scenes (Maharana, Hannan, and Bansal 2022; Pan et al. 2024; Rahman et al. 2023). Moreover, methods that generate all frames simultaneously, although maintaining coherence, often produce images that are very similar to each other (Zhou et al. 2024). Some approaches incorporate sketches or personalization to introduce control, but these methods often limit versatility (Cheng et al. 2024a; Wang et al. 2023). Although some approaches explored editing within the context of story visualization, their editing capabilities are limited to individual frames and do not support consistent, multi-frame modifications across the entire narrative which is crucial for story coherence (Cheng et al. 2024a,b). To address these limitations, we present `Plot'n Polish`, a novel zero-shot

| Methods/ task | Multi-turn Editing | Multi-frame Editing | Publicly Available |
|---|---|---|---|
| AutoStory | ✗ | ✗ | ✗ |
| TaleCrafter | ✗ | ✗ | ✗ |
| IntelligentGrimm | ✗ | ✗ | ✓ |
| StoryDiffusion | ✗ | ✗ | ✓ |
| ConsiStory | ✗ | ✗ | ✓ |
| Theatergen | ✓ | ✗ | ✓ |
| AutoStudio | ✓ | ✗ | ✓ |
| **Plot'n Polish (ours)** | ✓ | ✓ | ✓ |

Table 1: Plot'n Polish is the only story visualization method that enables multi-frame editing.


Figure 2: The story template is initially generated using an off-the-shelf T2I model, such as SDXL (top row). Our method edits these inconsistencies, transforming the template into a series of consistent story panels (bottom row).

framework that provides users with comprehensive control over creating and refining story visualizations while maintaining consistency across multiple frames. Our method enables users to make both fine-grained adjustments and substantial modifications, such as replacing a character or applying stylistic changes across all scenes (Fig. 1) without any training or fine-tuning. This flexibility allows for experimentation with different artistic styles and visual elements without the need to regenerate frames individually. Unlike previous methods that lack post-editing flexibility, our approach incorporates a novel editing mechanism that leverages inter-frame correspondences, ensuring edits are consistently applied across multiple frames. Our method supports both the generation of consistent story visuals and the seamless editing of generated frames, as well as user-provided story visuals, whether created by other methods or sourced from published books. By maintaining both narrative continuity and stylistic coherence throughout the sequence, `Plot'n Polish` empowers creators to craft, refine, and enhance their visual narratives. Our contributions can be summarized as follows:

- We introduce `Plot'n Polish`, a novel method that supports both initial story visualization and post generation modifications that ensures multi-frame consistency for both local and global edits.
- Our approach seamlessly integrates with existing workflows by supporting the editing of previously generated or user-provided story visuals, including real-world illustrations from published storybooks.
- Our method accommodates a wide range of editing tasks, from fine-grained adjustments to character transformations and personalization, empowering users to refine story visuals with precision, adapt elements to fit specific narratives, and seamlessly integrate customized details into the story frames.
- Through comprehensive experiments, we show that Plot'n Polish outperforms state-of-the-art story visualization and editing methods in terms of consistency, text alignment, and editing flexibility.

## Related Work

**Story Visualization** Recent work has explored the use of text-to-image models for story generation. (Pan et al. 2024; Liu et al. 2024; Rahman et al. 2023; Yang et al. 2024) introduce autoregressive models that require training on specific datasets, limiting their generalization capabilities and compromising image quality compared to the base model. (Gong et al. 2023; Wang et al. 2023; Avrahami et al. 2023) rely on LoRA (Hu et al. 2021) models, requiring finetuning for each character. (Jeong, Kwon, and Ye 2023) focuses on face editing using a personalization method. (Tao et al. 2024) generates multi-frame storyboards but lacks fine-grained editing capabilities. (Cheng et al. 2024a) introduces an additional U-Net (Ronneberger, Fischer, and Brox 2015) to improve identity consistency. While it addresses multi-turn editing, it does not ensure consistency across multiple edited images—an essential aspect for interactive storytelling. While methods such as (Liu et al. 2025; Zhou et al. 2024; Tewel et al. 2024) utilize attention sharing for story generation, they do not perform editing. Overall, our method is the only method that supports multi-frame editing in the story visualization context, allowing users to apply consistent edits across the entire story sequence.

**Image Editing with Diffusion Models.** Many diffusion-based editing approaches rely on text prompts to specify modifications. (Brooks, Holynski, and Efros 2022) enables text-driven image edits but often results in entangled edits, affecting unintended areas. Some works improve control over the editing process (Hertz et al. 2022; Zhang, Rao, and Agrawala 2023), such as ControlNet (Zhang, Rao, and Agrawala 2023), which uses conditional inputs to adjust specific image attributes. Similarly, (Valevski et al. 2022) ensures content preservation by fine-tuning the diffusion model on the input image. (Tumanyan et al. 2023) retrieves inversion noise and applies denoising for feature reconstruction. To enhance flexibility, recent approaches such as (Brack et al. 2023, 2024; Liu et al. 2022) decompose edits into multiple components, enabling finer control over modifications. However, most of these methods focus on editing single images, overlooking multi-frame consistency. In contrast, our method edits the same object across multiple frames simultaneously by leveraging attention sharing for localized masked-area editing.

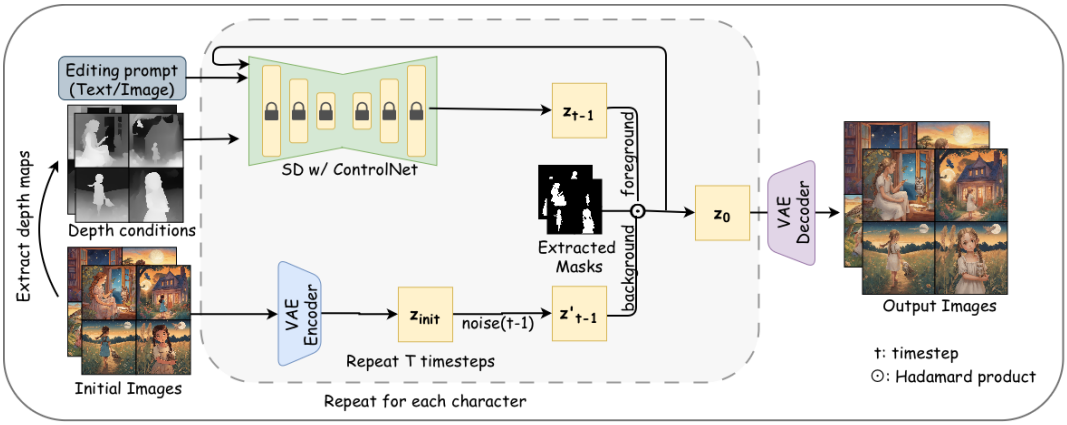


Figure 3: An overview of Plot'n Polish. Given story plot text and image prompts—either user-provided or LLM generated—the framework first creates initial story images and extracts their depth maps. Using ControlNet, it refines each frame over with editing prompts and mask-guided updates. The process is repeated for each character to ensure consistent story visuals.

**Grid Priors.** Prior works such as RAVE (Kara et al. 2024) and NeRFiller (Weber et al. 2023) leverage grid priors for consistency or scene completion. Unlike these, our method applies them for fine-grained, disentangled control. RAVE ensures temporal consistency for video edits but lacks disentangled editing, changes to a character can unintentionally affect the background. In contrast, our approach performs localized latent blending for isolated edits within story frames, enabling coherent narratives from scratch. StoryImager (Tao et al. 2024) also employs a grid layout but is limited by fixed frame counts and extensive training. Our method is training-free, allowing longer stories and on-the-fly editing. Similarly, ORACLE (Akdemir and Yanardag 2024) uses grids to train a LoRA (Hu et al. 2021) for identity consistency, whereas ours achieves zero-shot multi-character story generation and editing without fine-tuning.

## Methodology

We present Plot 'n Polish, an end-to-end interactive pipeline that provides users with comprehensive control over creating and refining story visualizations while maintaining consistency across multiple frames. Our framework is flexible enough to edit existing story frames provided by users or to help users generate new stories from simple ideas, such as 'Create me a story about a girl and a cat discovering their backyard.' For users looking to create story visualizations from scratch, we first employ a simple strategy by utilizing LLMs to generate story narratives (see Appendix). Alternatively, users can provide plot texts and image prompts for the story visualization.

Our novel multi-frame editing method is designed for two main purposes: 1) ensuring consistency in initial story visualizations and 2) performing coherent edits based on user-provided text prompts. Both tasks are integrated into a unified framework with slight task-specific adaptations. See Figure 3 for an overview of the pipeline.

### Story Visualization

Given a set of story prompts $\mathcal{P}$ either generated by an LLM or provided by the user, we first create an initial story template of $n$ frames denoted as $\mathcal{F} = \{F_1, F_2, \ldots, F_n\}$, corresponding to the prompt set $\mathcal{P}$, using an off-the-shelf T2I model such as SDXL (Podell et al. 2023). This approach ensures that the diversity of layouts reflects the capabilities of the original model, including backgrounds and objects as described by the text prompt. However, the generated template often lacks consistency, with characters and details varying significantly across frames (refer to Fig. 2 (top) where the girl character and her attire are inconsistent across different frames). To address this issue, we employ our novel multi-frame editing framework that refines the initial story template to achieve consistent character representation across frames (refer to Fig. 2 (bottom)) as described below. We iteratively refine each character for cross-frame consistency.

### Multi-Frame Editing

Given the initial story template $\mathcal{F}$, a concept $k$ to modify (such as 'boy') and an editing prompt $\mathcal{P}_{edit}$ such as 'a boy with short hair and a striped t-shirt', we first extract masks $\mathcal{M}_k = \{M_1, M_2, \ldots, M_n\}$ using object detection and semantic segmentation models (Cheng et al. 2024c; Xiong et al. 2023). After obtaining the masks, we generate consistent images that will align with the editing prompt $\mathcal{P}_{edit}$ to modify the masked regions of the original frames while ensuring visual consistency. To enforce consistency across multiple frames, we leverage the grid prior (Weber et al. 2023), which enables interaction of spatial information across multiple frames. This approach ensures that modifications remain coherent throughout the sequence while allowing the model to leverage spatial relationships between frames. In addition, we incorporate ControlNet (Zhang and Agrawala 2023) to preserve the structural integrity of the original image while guiding modifications within the specified regions.

Given $n$ frames, we partition them into groups of size $\gamma = \phi \times \beta$ and arrange them into a rectangular grid. The grid

representation is defined as follows:

$$Grid(X_1, X_2, ....X_\gamma) = \begin{bmatrix} X_1 & \dots & X_\beta \\ \vdots & \ddots & \\ X_{(\phi-1)*\beta+1} & & X_\gamma \end{bmatrix}.$$

We form grids $z_{\text{grid}}$ corresponding to the latent representations, $m_{\text{grid}}$ to the masks, $d_{\text{grid}}$ to the depth conditions, and $F_{\text{grid}}$ to the original frames, ensuring structured processing across multiple frames. At each diffusion timestep $t \in \{T, T-1, \dots, 1\}$, we update the latent representation for each grid as follows:

$$z_{\text{grid},t-1} \leftarrow \epsilon_\theta(z_{\text{grid},t}, d_{\text{grid}}, t, \mathcal{P}_{edit}).$$

To ensure that all frames interact with each other and to prevent isolated inconsistencies, we reform the grids at each timestep by randomly regrouping the frames. This process reduces memory overhead while providing cross-frame consistency.

To prevent modifications from unintentionally altering unmasked regions, we apply latent blending after each denoising step. Inspired by (Avrahami, Fried, and Lischinski 2022), this approach ensures that the edited frames retain the original background details while applying changes only to the masked regions. We incorporate a secondary noised latent representation derived from the original grid, defined as:

$$z'_{\text{grid},t-1} = \sqrt{\alpha_{t-1}}\mathcal{E}(F_{\text{grid}}) + \sqrt{1-\alpha_{t-1}}\epsilon,$$

where $\mathcal{E}$ denotes the encoder, $\alpha_t$ controls the noise schedule, and $\epsilon \sim \mathcal{N}(0, I)$ is a sampled Gaussian noise term. We then apply the latent blending process:

$$z_{\text{grid},t-1} \leftarrow z_{\text{grid},t-1} \odot m + z'_{\text{grid},t-1} \odot (1-m),$$

where $m$ is the resized version of $M_{\text{grid}}$ to match the latent resolution, and $\odot$ denotes element-wise multiplication. For global edits such as style modifications, we omit this blending step, allowing changes to propagate across the entire image.

Once the denoising process is complete, we decode the final latent grid back into the pixel space to obtain the final images. This reconstruction step is performed as:

$$x_{\text{grid},0} = \mathbf{D}(z_{\text{grid},0}),$$

where $\mathbf{D}$ represents the decoder of the diffusion model. By incorporating depth conditions and latent blending, we effectively balance consistency and diversity, ensuring that the final frames maintain their structural coherence while allowing for expressive modifications.

**Personalization.** Our method enables personalization by allowing users to customize generated characters, animals, or objects using either a pre-trained LoRA (Hu et al. 2021) model or a single reference image. We achieve this by incorporating the IP-Adapter (Ye et al. 2023) into our pipeline, specifically by loading its cross-attention layers. This integration allows the model to process image-based prompts, seamlessly embedding personalized elements into the generation process.

# Experiments

We evaluated Plot'n Polish against both state-of-the-art story visualization methods (Zhou et al. 2024; Tewel et al. 2024; Cheng et al. 2024a; Liu et al. 2024) and editing methods (Brooks, Holynski, and Efros 2022; Brack et al. 2024; Tumanyan et al. 2023) through a series of qualitative and quantitative experiments. The appendix is available on our project page.

## Experimental Setup

All template images are generated using SDXL, while image editing is performed with Stable Diffusion (SD) 1.5. A 3×3 grid is employed to generate story panels. For editing, we apply depth-conditioned ControlNet, using a depth condition of 0.4 for local edits and 1.0 for global edits. Experiments are conducted on a single NVIDIA A40 GPU.

## Qualitative Experiments

**Qualitative Results.** Fig. 4 presents a diverse range of qualitative experiments demonstrating the versatility of our method for consistent story generation and editing based on simple text prompts. Our method is able to maintain characters' consistency, including retaining their attire, such as hats (Fig. 4 (a)) or dresses (Fig. 4 (c)), across multiple panels. Similarly, smaller objects such as a fish (Fig. 1) or a parrot (Fig. 4 (b)) are accurately maintained throughout the story, demonstrating fine-grained visual and contextual coherence. The story texts and prompts are provided in the Appendix.

Moreover, our approach supports fine-grained edits, such as modifying specific object attributes like changing hair to 'glitter hair' (Fig. 1), removing small objects like bowtie (Fig. 4 (a)) or altering the color of a t-shirt to red (Fig. 4 (b)). Additionally, our method enables character transformations, such as replacing a fish with a crystal (Fig. 1) or changing a parrot into a lizard (Fig. 4 (b)). These capabilities are crucial for creative applications, where multiple iterative edits and refinements are often required to achieve the desired narrative or visual composition. Our method can also perform stylistic transformations, allowing users to apply global changes across all story panels, such as rendering them in the style of 'Van Gogh' or 'Day of the Dead' (Fig. 4 (c)). This flexibility is particularly valuable in the creative process: users can initially focus on designing the story's characters and elements, then experiment with different artistic styles during later stages of refinement. Additionally, our method supports personalization. Users can provide a single reference image of a desired concept or a LoRA model, enabling integration of custom elements into the story (Fig. 4 (d)). The original versions of these personalized stories can be found in the Appendix.

Furthermore, our method can seamlessly edit user-provided story frames, whether from existing books or generated by other methods, without requiring any special adjustments. To demonstrate its ability to modify existing visual stories, we edited frames from two different stories sourced from the Gutenberg[1] in Fig 6. Our approach successfully transforms characters, such as changing duck to a

[1]https://www.gutenberg.org


Figure 4: Qualitative results for Plot'n Polish. Our results demonstrate that Plot'n Polish excels in producing consistent visual narratives and allows for a wide range of successful edits including localized edits, entity replacements, and personalization.


Figure 5: Qualitative comparison of our method with state-of-the-art story visualization methods. Our method outperforms competitors by maintaining consistent visual elements, such as attire and character features, across all panels, ensuring narrative coherence. In contrast, existing methods struggle with inconsistencies and blending errors.

rooster (Fig. 6 left), as well as performs fine-grained edits like changing a coat to a trench coat or altering hair color and style (Fig. 6 right) consistently. These results highlight the robustness and versatility of our approach, showcasing its ability to perform detailed edits, handle diverse stylistic changes, and support personalized storytelling; essential for enhancing creative workflows. Please see the Appendix for additional story visualizations, including longer stories.

**Qualitative Comparison.** We compare our method to both

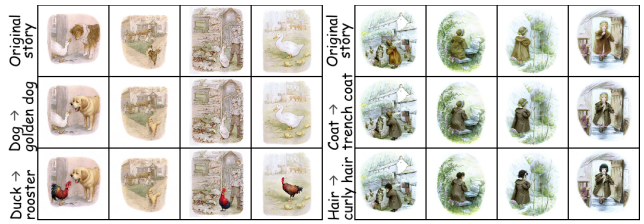


Figure 6: Our method can edit existing stories, ranging from transformations such as transforming characters (e.g., a duck to a rooster) to fine-grained modifications like altering clothing (e.g., a coat to a trench coat) or hairstyle.

state-of-the-art story visualization methods (Zhou et al. 2024; Tewel et al. 2024; Cheng et al. 2024a; Liu et al. 2024) and editing methods (Brooks, Holynski, and Efros 2022; Brack et al. 2024; Tumanyan et al. 2023). Note that we were not able to compare with (Gong et al. 2023; Tao et al. 2024) since their code is not publicly available. We first evaluate story visualization consistency in Fig. 5. Our method excels at maintaining consistent visual elements, such as preserving both girl and Captain characters' attire across all panels. In contrast, other methods struggle to achieve the same level of consistency or accurate character depiction. For instance, StoryDiffusion (Zhou et al. 2024) fails to maintain consistent attire for both characters(see Captain character). Consistory (Tewel et al. 2024) maintains some level of consistency but omits the Captain character in the third and sixth panels while introducing additional characters in the first and fifth panels, deviating from the intended narrative. AutoStudio (Cheng et al. 2024a) frequently changes character attire and generating blended visuals (e.g. Captain is wearing a

| | Method | CLIP-I↑ | CLIP-T↑ | DINO↑ | LPIPS↓ | User-I↑ | User-T↑ | User-D ↑ | Time(s)↓ |
|---|---|---|---|---|---|---|---|---|---|
| **Story Generation** | StoryDiffusion | 0.78 ± 0.05 | 0.32 ± 0.05 | 0.41 ± 0.10 | 0.49 ± 0.04 | 2.55 ± 1.29 | 2.44 ± 1.27 | N/A | **9s** |
| | Intelligent Grimm | 0.78 ± 0.05 | 0.30 ± 0.02 | **0.52 ± 0.11** | 0.63 ± 0.03 | 2.09 ± 1.13 | 2.08 ± 1.16 | N/A | 15s |
| | Consistory | **0.83 ± 0.04** | 0.34 ± 0.02 | 0.49 ± 0.11 | 0.51 ± 0.03 | 2.70 ± 1.47 | 2.64 ± 1.52 | N/A | 17s |
| | AutoStudio | 0.78 ± 0.05 | 0.33 ± 0.02 | 0.38 ± 0.18 | 0.48 ± 0.18 | 2.73 ± 1.29 | 2.73 ± 1.30 | N/A | 11s |
| | Plot'n Polish (Ours) | 0.79 ± 0.05 | **0.36 ± 0.02** | 0.41 ± 0.12 | **0.47 ± 0.06** | **2.90 ± 1.30** | **2.82 ± 1.30** | N/A | 11s |
| **Story Editing** | InstructPix2Pix | 0.89 ± 0.12 | 0.31 ± 0.08 | 0.77 ± 0.24 | 0.21 ± 0.19 | 2.37 ± 1.26 | 2.23 ± 1.21 | 1.65 ± 0.98 | 9s |
| | LEDITS++ | 0.76 ± 0.07 | 0.29 ± 0.06 | 0.50 ± 0.16 | 0.24 ± 0.05 | 2.35 ± 1.21 | 2.62 ± 1.32 | 2.49 ± 1.35 | **5s** |
| | Plug-and-Play | 0.82 ± 0.09 | 0.30 ± 0.06 | 0.67 ± 0.16 | 0.30 ± 0.08 | 2.13 ± 1.21 | 2.41 ± 1.30 | 2.46 ± 1.25 | 250s |
| | Plot'n Polish (Ours) | **0.93 ± 0.06** | **0.33 ± 0.07** | **0.88 ± 0.17** | **0.10 ± 0.07** | **3.99 ± 1.07** | **4.08 ± 1.06** | **4.17 ± 1.04** | 9s |

Table 2: Quantitative results for story generation and editing tasks, evaluated using CLIP-I, CLIP-T, DINO, and LPIPS metrics, along with user study results (1-5 scale) for image consistency (User-I), text alignment (User-T), and disentanglement (User-D).

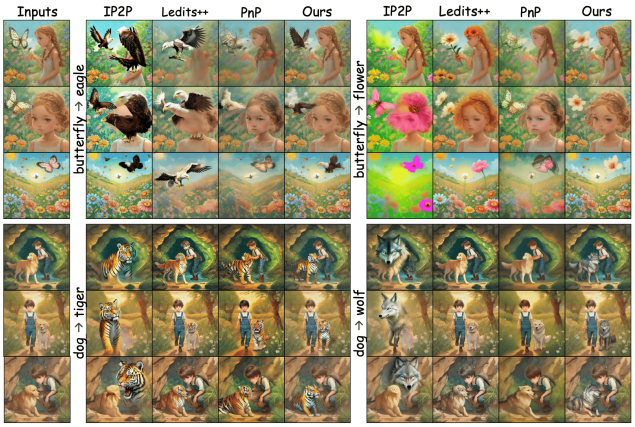


Figure 7: Qualitative comparison of editing methods. Our approach produces more consistent and disentangled edits, editing accurately while preserving the character details. In contrast, LEDITS++ shows blending artifacts, Plug-and-Play (PnP) often fails to generate target objects, and InstructPix2Pix (IP2P) frequently alters unintended regions.

skirt). Finally, IntelligentGrimm (Liu et al. 2024) is heavily constrained by its reliance on a specific dataset, limiting the style and flexibility. Additionally, (Liu et al. 2024) often fails to depict key story elements accurately (e.g. omitting the Captain character entirely in the second panel). In contrast, our method consistently maintains coherent visuals. See Appendix for more qualitative comparisons.

Although the generation code for AutoStudio (Cheng et al. 2024a) is publicly available, the editing code is not, preventing a quantitative comparison. However, see Appendix for a qualitative comparison between their method and ours, using images taken from the original paper.

In Fig. 7, we compare our method against state-of-the-art editing methods (Brooks, Holynski, and Efros 2022; Brack et al. 2024; Tumanyan et al. 2023). Each row represents a different edit scenario, including modifications like replacing a butterfly with an eagle or a flower and transforming a dog into a tiger or wolf. InstructPix2Pix (Brooks, Holynski, and Efros 2022) introduces unintended changes beyond the target object, altering key aspects of the scene, such as modifying the girl's appearance instead of focusing solely on the intended edit. Ledits++ (Brack et al. 2024) struggles with blending, e.g., merging the dog and tiger unnaturally or distorting the girl's face during transformation. P2P (Tumanyan et al. 2023) fails to generate the requested objects in some cases, such as being unable to synthesize an eagle or a flower as intended. In contrast, our method maintains both visual consistency (e.g., generating the exact same flower across frames) and semantic accuracy, and only the specified modifications are applied while keeping the rest of the scene unchanged.

## Quantitative Experiments

We conducted quantitative experiments by generating 200 stories, each consisting of 9 frames, resulting in a total of 1800 frames. The initial stories were generated using GPT-4 (Achiam et al. 2023), featuring diverse characters. Table 2 reports several key metrics: image similarity (CLIP-I (Radford et al. 2021), DINO (Caron et al. 2021), LPIPS (Zhang et al. 2018)) and text similarity (CLIP-T (Radford et al. 2021)). See Appendix for experiment setup.

**Story Visualization** For story visualization, our method outperforms other approaches in CLIP-T, indicating its superior ability to generate visuals that align closely with the provided text prompts. Additionally, our method achieves higher CLIP-I scores than most competitors, demonstrating its ability to maintain consistency across story panels—a crucial aspect of coherent storytelling. An exception is ConsiStory (Tewel et al. 2024), which attains a high CLIP-I score of 0.83. However, this comes at a significant cost: as shown in Fig. 5, ConsiStory's generated panels fail to accurately depict the intended characters, either omitting them or introducing additional ones (see Appendix for more examples), diverging from the intended narrative. Furthermore, our method outperforms in LPIPS and DINO metrics, reinforcing that we generate consistent story panels.

**Story Editing** For editing task (see Table 2) our method outperforms in both preserving the original image and aligning with the given edit prompt. It achieves the highest CLIP-I and DINO scores, indicating strong consistency with the original image, and the lowest LPIPS, demonstrating minimal unintended changes. Additionally, it achieves the best CLIP-T scores, confirming our method's superior ability to


Figure 8: We ablate the following components: grid size, grid priors, and latent blending. See red-circled areas for inconsistencies in characters and background artifacts.

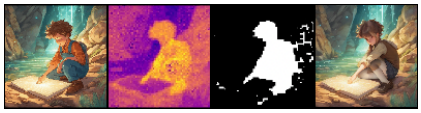

Figure 9: Example of image editing using attention mask.

apply edits while respecting to the original image.

**User Study.** We employed 50 participants from Prolific.com across 60 story frames for each method for user assessment. Each participant was shown story panels where the corresponding narrative was included, and asked to evaluate two key aspects of the story visualization: 1) Alignment with the narrative 2) Consistency across panels. Table 2 presents the results, comparing our method to state-of-the-art approaches. Our method was rated highest for both narrative alignment and consistency, demonstrating its ability to generate visuals that closely follow the story text while maintaining coherent character and object depictions. For editing, each participant was presented with three images alongside their corresponding edits and asked to assess three key aspects: 1) Alignment with the edit prompt , 2) Disentanglement and 3) Consistency. For both user studies, we used a rating scale from 1 (Very Bad) to 5 (Very Good). Our method received the highest ratings across all aspects, demonstrating its ability to perform consistent and disentangled editing.

**Story Generation and Editing Time.** Our method exhibits high efficiency, generating initial frames in just 2 seconds and an editing speed of 9s per frame, totaling to 11s. As shown in Table 2, our approach performs comparably to state-of-the-art story generation and editing methods while maintaining a balance between speed and quality.

**Ablation Study.** Our ablation study investigates the impact of different consistency mechanisms on the generated stories, as illustrated in Figure 8 (Refer to an additional example in the Appendix). Increasing the grid size enhances frame-to-frame interaction, which in turn improves visual consistency. For instance, the 3×3 grid exhibits better character continuity compared to the 2×2 variant. When grid priors are removed, meaning each frame is edited independently, the model lacks any explicit mechanism for enforcing consistency, often resulting in noticeable variations in character appearance. Likewise, omitting latent blending in favor of a copy-and-paste approach introduces visible artifacts between frames. These findings underscore the importance of both grid structure and latent blending in maintaining coherent visual narratives.


Figure 10: Failure case: Due to overlapping mask detection, the middle figure incorrectly includes additional subjects.

Furthermore, our method is inherently flexible and can accommodate alternative masking strategies. To test this, we conduct an experiment using attention-based masks derived from the model's attention maps, replacing explicit segmentation. As shown in Fig. 9, while segmentation masks offer more precise and controllable edits, attention-derived masks still yield compelling results. This demonstrates the robustness of our approach across different sources of masks.

We also verify that our method generalizes effectively to high-resolution diffusion models such as SDXL (Podell et al. 2023) and Flux (Labs 2024), with qualitative results provided in the Appendix.

## Limitation and Societal Impact

Our method leverages segmentation models to detect masks for characters or objects targeted for editing. As a result, the quality of our edits is influenced by the performance of these models. For instance, in cases where multiple overlapping objects exist, such as two reindeer in Fig. 10, our method may edit the unintended subjects. Although our method relies on segmentation model (Xiong et al. 2023), it can efficiently process an entire batch of 10 images in just two seconds, making them a more practical and scalable solution.

## Conclusion

In this paper, we introduce Plot'n Polish, a novel story-visualization framework that addresses key limitations of prior methods by enabling multi-frame editing, ensuring cross-frame consistency, and supporting both fine-grained and large-scale modifications. Unlike prior approaches, our method allows users to edit entire story sequences while preserving coherence, making changes such as altering attire, adding accessories, replacing characters, or transforming visual styles. Edits are applied consistently across frames without requiring full regeneration, enhancing creative control. Additionally, personalization features, including user-provided concepts, further expand its versatility. Future work will explore interactive branching narratives to support more complex story dynamics.

## References


Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akdemir, K.; and Yanardag, P. 2024. Oracle: Leveraging mutual information for consistent character generation with loras in diffusion models. *arXiv preprint arXiv:2406.02820*.

Avrahami, O.; Fried, O.; and Lischinski, D. 2022. Blended Latent Diffusion. *arXiv preprint arXiv:2206.02779*.

Avrahami, O.; Hertz, A.; Vinker, Y.; Arar, M.; Fruchter, S.; Fried, O.; Cohen-Or, D.; and Lischinski, D. 2023. The Chosen One: Consistent Characters in Text-to-Image Diffusion Models. *arXiv preprint arXiv:2311.10093*.

Brack, M.; Friedrich, F.; Hintersdorf, D.; Struppek, L.; Schramowski, P.; and Kersting, K. 2023. Sega: Instructing diffusion using semantic dimensions. *arXiv preprint arXiv:2301.12247*.

Brack, M.; Friedrich, F.; Kornmeier, K.; Tsaban, L.; Schramowski, P.; Kersting, K.; and Passos, A. 2024. LEDITS++: Limitless Image Editing using Text-to-Image Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2022. InstructPix2Pix: Learning to Follow Image Editing Instructions. *arXiv preprint arXiv:2211.09800*.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9650–9660.

Cheng, J.; Lu, X.; Li, H.; Zai, K. L.; Yin, B.; Cheng, Y.; Yan, Y.; and Liang, X. 2024a. AutoStudio: Crafting Consistent Subjects in Multi-turn Interactive Image Generation. *arXiv preprint arXiv:2406.01388*.

Cheng, J.; Yin, B.; Cai, K.; Huang, M.; Li, H.; He, Y.; Lu, X.; Li, Y.; Li, Y.; Cheng, Y.; et al. 2024b. TheaterGen: Character Management with LLM for Consistent Multi-turn Image Generation. *arXiv preprint arXiv:2404.18919*.

Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; and Shan, Y. 2024c. YOLO-World: Real-Time Open-Vocabulary Object Detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Gong, Y.; Pang, Y.; Cun, X.; Xia, M.; He, Y.; Chen, H.; Wang, L.; Zhang, Y.; Wang, X.; Shan, Y.; et al. 2023. Talecrafter: Interactive story visualization with multiple characters. *arXiv preprint arXiv:2305.18247*.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jeong, H.; Kwon, G.; and Ye, J. C. 2023. Zero-shot Generation of Coherent Storybook from Plain Text Story using Diffusion Models. arXiv:2302.03900.

Kara, O.; Kurtkaya, B.; Yesiltepe, H.; Rehg, J. M.; and Yanardag, P. 2024. RAVE: Randomized Noise Shuffling for Fast and Consistent Video Editing with Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Labs, B. F. 2024. FLUX. https://github.com/black-forest-labs/flux.

Liu, C.; Wu, H.; Zhong, Y.; Zhang, X.; Wang, Y.; and Xie, W. 2024. Intelligent Grimm - Open-ended Visual Storytelling via Latent Diffusion Models. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6190–6200.

Liu, N.; Li, S.; Du, Y.; Torralba, A.; and Tenenbaum, J. B. 2022. Compositional Visual Generation with Composable Diffusion Models. *arXiv preprint arXiv:2206.01714*.

Liu, T.; Wang, K.; Li, S.; van de Weijer, J.; Khan, F. S.; Yang, S.; Wang, Y.; Yang, J.; and Cheng, M.-M. 2025. One-Prompt-One-Story: Free-Lunch Consistent Text-to-Image Generation Using a Single Prompt. arXiv:2501.13554.

Maharana, A.; Hannan, D.; and Bansal, M. 2022. Storydall-e: Adapting pretrained text-to-image transformers for story continuation. In *European Conference on Computer Vision*, 70–87. Springer.

Pan, X.; Qin, P.; Li, Y.; Xue, H.; and Chen, W. 2024. Synthesizing coherent story with auto-regressive latent diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2920–2930.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. arXiv:2307.01952.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rahman, T.; Lee, H.-Y.; Ren, J.; Tulyakov, S.; Mahajan, S.; and Sigal, L. 2023. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2493–2502.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Tao, M.; Bao, B.-K.; Tang, H.; Wang, Y.; and Xu, C. 2024. StoryImager: A Unified and Efficient Framework for Coherent Story Visualization and Completion. *arXiv preprint arXiv:2404.05979*.

Tewel, Y.; Kaduri, O.; Gal, R.; Kasten, Y.; Wolf, L.; Chechik, G.; and Atzmon, Y. 2024. Training-Free Consistent Text-to-Image Generation. arXiv:2402.03286.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2023. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1921–1930.

Valevski, D.; Kalman, M.; Molad, E.; Segalis, E.; Matias, Y.; and Leviathan, Y. 2022. UniTune: Text-Driven Image Editing by Fine Tuning a Diffusion Model on a Single Image. *ACM Transactions on Graphics (TOG)*, 42: 1 – 10.

Wang, W.; Zhao, C.; Chen, H.; Chen, Z.; Zheng, K.; and Shen, C. 2023. Autostory: Generating diverse storytelling images with minimal human effort. *arXiv preprint arXiv:2311.11243*.

Weber, E.; Hołyński, A.; Jampani, V.; Saxena, S.; Snavely, N.; Kar, A.; and Kanazawa, A. 2023. NeRFiller: Completing Scenes via Generative 3D Inpainting. arXiv:2312.04560.

Xiong, Y.; Varadarajan, B.; Wu, L.; Xiang, X.; Xiao, F.; Zhu, C.; Dai, X.; Wang, D.; Sun, F.; Iandola, F.; Krishnamoorthi, R.; and Chandra, V. 2023. EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything. arXiv:2312.00863.

Yang, S.; Ge, Y.; Li, Y.; Chen, Y.; Ge, Y.; Shan, Y.; and Chen, Y. 2024. SEED-Story: Multimodal Long Story Generation with Large Language Model. *arXiv preprint arXiv:2407.08683*.

Ye, H.; Zhang, J.; Liu, S.; Han, X.; and Yang, W. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models.

Zhang, L.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Zhou, Y.; Zhou, D.; Cheng, M.-M.; Feng, J.; and Hou, Q. 2024. StoryDiffusion: Consistent Self-Attention for Long-Range Image and Video Generation. *arXiv preprint arXiv:2405.01434*.