StoryBooth: Consistent Personalized Comic Generation with Diffusion Models

CS492 Visual Generation Contest, Team 9

Jaeyoung Shin
20230904
KAIST

Jaewoo Yu
20210400
KAIST

Sangoh Kim
20220112
KAIST

Figure 1. **Final Visual Content.** Given a few reference images, we generate comic images preserving characters identity consistently by utilizing LoRA and attention manipulation.

1. Introduction

Generating personalized visual narratives with consistent character identity across multiple images remains a significant challenge. While text-to-image diffusion models such as Stable Diffusion XL (SDXL) [5] have achieved remarkable progress in generating high-quality single images, maintaining character consistency throughout sequential comic panels requires addressing two fundamental problems: (1) personalized subject representation and (2) cross-

image identity preservation.

Personalization techniques, including DreamBooth [6] and Textual Inversion [1], enable users to inject specific subjects into pretrained diffusion models using only a few reference images. DreamBooth fine-tunes the model weights to associate a unique identifier token with the target subject, while Textual Inversion learns a new embedding in the text encoder's vocabulary space. However, these methods alone cannot guarantee that same context preserved in multiple images.

To address this challenge, StoryDiffusion [9] introduces Consistent Self-Attention, a training-free mechanism that enables character identity preservation across long-range image sequences. By sharing self-attention features between reference identity images and newly generated panels, StoryDiffusion achieves remarkable consistency without additional fine-tuning.

In this work, we present **StoryBooth**, an integrated framework that combines the strengths of LoRA tuning with consistent self-attention for personalized comic generation. Our approach leverages DreamBooth with Low-Rank Adaptation (LoRA) [3] for efficient personalization, while employing StoryDiffusion's consistent self-attention mechanism to maintain character identity across comic panels. Additionally, we explore the impact of alternative timestep sampling distributions during DreamBooth training, specifically investigating log-normal sampling [4] as a replacement for uniform timestep selection, which improves the personalization ability in diffusion models.

Our contributions are as follows:

- We propose StoryBooth, a unified pipeline that integrates DreamBooth-LoRA fine-tuning with StoryDiffusion's consistent self-attention for personalized comic generation.
- We investigate the effect of log-normal timestep sampling during personalization training and demonstrate its benefits for subject fidelity.

2. Brief Description of Visual Content

We made nine cuts of cartoon starring Nupjuki, KAIST's official mascot.

2.1. Story

Long ago, one quiet day, Nupjuki was sitting in a small café, reading a newspaper. As he flipped through its pages, he happened to discover a tale about a mysterious "House of Treasure." His heart fluttered with curiosity, and without a moment's delay, he rose from his seat and hurried out of the café, racing toward the distant jungle. Deep within the thick, shadowy forest, Nupjuki wandered in search of the legendary house. But all of a sudden, a giant spider appeared before him. Startled and trembling with fear, Nupjuki turned and ran as fast as his legs would carry him, not daring to look back. He kept running until he was completely exhausted. Just then, he noticed a small cabin hidden among the trees, softly glowing with the light of fireflies. Drawn to its warm shimmer, he pushed the door open—and to his astonishment, the room was filled with gold and countless treasures. From that day on, Nupjuki lived a life of comfort and happiness, all thanks to the treasure he had found.

3. Technical Details

Our StoryBooth framework builds upon two key components: DreamBooth-LoRA for personalization and StoryDiffusion's Consistent Self-Attention for cross-image consistency. Overall framework is described in Fig. 2.

3.1. DreamBooth with LoRA

3.1.1. DreamBooth

DreamBooth [6] enables personalization by fine-tuning the pretrained diffusion model on a small set of subject images (typically 3-5 images). Given N reference images $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of a specific subject, DreamBooth associates the subject with a unique identifier token V^* by minimizing the diffusion training objective:

$$\mathcal{L}_{\text{DB}} = \mathbb{E}_{\mathbf{z},\mathbf{c},\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c})\|_2^2\right] \tag{1}$$

where $\mathbf{z} = \mathcal{E}(\mathbf{x})$ is the encoded latent, $\mathbf{c}$ contains the unique identifier, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the added noise, and t is a randomly sampled timestep.

Our framework adopted DreamBooth technique with LoRA [3] due to computational efficiency.

3.1.2. Log-Normal Timestep Sampling

Standard diffusion training uniformly samples timesteps $t \sim \mathcal{U}(0, T-1)$. However, recent work [4, 7] suggests that certain timestep ranges contribute more significantly to learning. We investigate log-normal timestep sampling as an alternative distribution:

$$t = \left\lfloor \frac{x}{1+x} \cdot (T-1) \right\rfloor, \quad \text{where } x \sim \text{LogNormal}(\mu, \sigma) \tag{2}$$

Log-Normal distribution makes training to focus on later timesteps which semantic information is primarily encoded. Thereby it prevents overfitting and injects fine-detail of subjects to the model effectively.

3.2. Consistent Self-Attention

3.2.1. Identity Feature Bank

StoryDiffusion [9] introduces Consistent Self-Attention, a training-free mechanism that preserves character identity across multiple generated images. The key insight is to

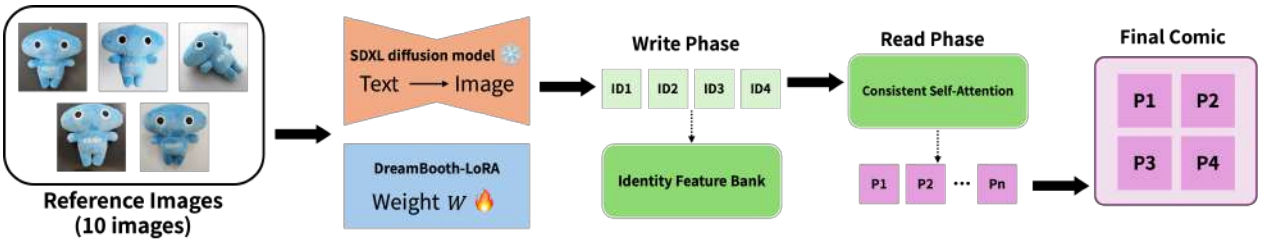

Figure 2. Overall framework of StoryBooth

maintain a shared feature bank that stores self-attention features from reference identity images and reuses them when generating new panels.

During generation, we first produce M identity reference images. These images establish the visual identity. For each subsequent panel generation, the self-attention layers access features from both the current image being generated and the stored identity features.

Specifically, at diffusion step t and attention layer l, we maintain an identity bank:

$$\mathcal{B}_t^l = \{\mathbf{h}_{1,t}^l, \mathbf{h}_{2,t}^l, \ldots, \mathbf{h}_{M,t}^l\} \tag{3}$$

where $\mathbf{h}_{i,t}^l \in \mathbb{R}^{N \times D}$ represents the hidden states from identity image i at timestep t and layer l, with N being the number of spatial tokens and D the hidden dimension.

3.2.2. Two-Phase Generation Process

The generation process consists of two phases:

Write Phase: Generate M identity reference images and store their self-attention features at each timestep and layer into the identity bank $\mathcal{B}_t^l$.

Read Phase: For each new comic panel, augment the self-attention input by concatenating the current image's features with stored identity features. Given hidden states $\mathbf{h}_{\text{new}}$ for the new image, we compute:

$$\mathbf{h}_{\text{concat}} = [\mathbf{h}_{1,t}^l, \mathbf{h}_{\text{new}}, \mathbf{h}_{2,t}^l, \ldots, \mathbf{h}_{M,t}^l] \tag{4}$$

The attention computation then operates on this concatenated representation:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_Q \mathbf{h}_{\text{concat}}, \\ \mathbf{K} &= \mathbf{W}_K \mathbf{h}_{\text{concat}}, \\ \mathbf{V} &= \mathbf{W}_V \mathbf{h}_{\text{concat}}, \end{aligned} \tag{5}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}.$$

where $\mathbf{M}$ is a spatial attention mask that controls which tokens can attend to which identity features.

3.2.3. Resolution-Specific Attention Masks

The attention mask $\mathbf{M}$ is constructed differently for different spatial resolutions in the UNet. For 32×32 resolution features (1024 tokens) and 64×64 resolution features (4096 tokens), we apply different consistency strengths s_{32} and s_{64} respectively. The mask ensures that:

- Identity images attend only to themselves and other identity images
- New panels attend to both identity features and their own features
- Attention weights are modulated by resolution-specific strength parameters

3.2.4. Stochastic Consistency

To maintain generation diversity while preserving identity, consistent self-attention is applied stochastically. For diffusion steps $t < 5$, we use standard self-attention to establish basic image structure. For $t \geq 5$, consistent self-attention is applied with probability:

$$p_{\text{consistent}} = \begin{cases} 0.7 & \text{if } t < 20 \\ 0.9 & \text{if } t \geq 20 \end{cases} \tag{6}$$

This scheduling allows early steps to explore diverse compositions while later steps enforce strong identity consistency.

3.3. StoryBooth: Integrated Pipeline

Our complete StoryBooth pipeline combines DreamBooth-LoRA and Consistent Self-Attention:

1. **Personalization:** Fine-tune SDXL with DreamBooth-LoRA on 10 reference images of the target character using the unique identifier token "a blue sks plush". We train for 1450 steps with learning rate 1×10^{-4} and LoRA rank $r = 4$.
2. **Identity Generation:** Load the personalized LoRA weights and generate $M = 2$ identity reference images. Store self-attention features from these images in the identity bank.
3. **Sequential Comic Generation:** For each comic panel with prompt $\mathbf{c}_i$, generate the image while accessing stored identity features through consistent self-attention.

This integrated approach ensures that the character is both personalized to match the reference images (via DreamBooth-LoRA) and consistent across all generated

panels (via Consistent Self-Attention). The LoRA weights provide subject-specific appearance details, while the attention mechanism enforces identity, context consistency across the narrative sequence.

3.4. Implementation Details

3.4.1. Dataset

Figure 3. Several samples from our dataset

We took 10 pictures of Nupjuki plush toy bought from KAIST Brandshop and built our own dataset for LoRA tuning.

3.4.2. StoryBooth (Our framework)

We use `sks` as our unique identifier. Also, we apply LoRA to the query, key, value, and output projection layers in the UNet's self-attention modules: `to_q`, `to_k`, `to_v`, and `to_out.0`.

For log-normal timestep sampling, we use hyperparameters $\mu = -0.3$ and $\sigma = 1.6$ which biases sampling toward mid-to-high noise levels where semantic information is primarily encoded.

We downloaded the SDXL checkpoint from Huggingface. Since SDXL's default VAE outputs `NaN` during LoRA tuning in fp16 precision, we use fixed version from here.

We use CFG [2] with guidance scale 5.0. Also we adopt positive and negative prompt used in StoryDiffusion [9].

4. Discussion of Artistic Aspects

Our visual content (Figure 1) preserves the appearance of Nupjuki and consistent with the story at the same time. To elaborate, when we compare Figure 1 and Figure 3, it's obvious that 'KAIST' logo reconstructed clearly and overall appearance are well preserved. Backgrounds are also consistent to the prompt and diverse.

5. How to Reproduce Experiments

5.1. Preparing Codebase

You can either unzip the file we submitted or clone from our repository.

```
git clone https://github.com/sangohkim/CS492-
    Visual_Content_Generation.git
```

5.2. Python Environment

We've listed all required packages at `requirements.yaml`. If some packages raise errors (it might occur due to some differences in overall environment), please download them manually.

```
conda env create -f requirements.yaml
```

5.3. SDXL Checkpoints

```
hf download stabilityai/stable-diffusion-xl-base
    -1.0
```

Due to the issue mentioned at Section 3.4.2, you need to download another VAE for LoRA tuning.

```
hf download madebyollin/sdxl-vae-fp16-fix
```

5.4. Training

Expected running time is about 2 hours.

You can follow below commands to start training. Dataset is already included in the zip file.

```
conda activate sdxl
cd dreambooth
bash scripts/train_lognormal.sh
```

Trained LoRA checkpoints are already provided at `dreambooth/results`.

Currently, at `scripts/train_lognormal.sh`, `OUTPUT_ROOT` is modified to a different path not to overwrite the existing LoRA checkpoints since this is only for verification. If you want to train it from scratch and use your trained checkpoints for generation, you can uncomment the original `OUTPUT_ROOT` in `scripts/train_lognormal.sh`. Also, please make sure to train at least 1450 steps since our results are based on that checkpoint. Our implementation is built on the Diffusers library [8].

5.5. Inference

Expected running time is about 20 minutes. You can generate images with below commands.

```
bash run_inference_final.sh
```

References

[1] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[3] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022. 2

[5] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 1

[6] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 2

[7] Vera Soboleva, Aibek Alanov, Andrey Kuznetsov, and Konstantin Sobolev. T-lora: Single image diffusion model customization without overfitting. *arXiv preprint arXiv:2507.05964*, 2025. 2

[8] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4

[9] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *Advances in Neural Information Processing Systems*, 37:110315–110340, 2024. 2, 4