# 基于 StoryDiffusion 模型的复现与改进

**摘要**

在当前基于扩散的生成模型中，保持一系列生成图像（尤其是包含复杂细节和主体的图像）之间的一致性颇具挑战。本文提出一种新的自注意力计算方式——一致自注意力（Consistent Self-Attention），能显著提升生成图像间的一致性，且以零样本方式增强现有的预训练扩散模型。为将此方法扩展到长视频生成，引入了语义空间时间运动预测模块（Semantic Motion Predictor），该模块在语义空间中预测两图像间的运动条件，能将生成的图像序列转换为过渡平滑且主体一致的视频，在长视频生成中表现尤为突出。结合这两个模块的 StoryDiffusion 框架，能够用一致的图像或视频来描绘基于文本的故事，涵盖丰富内容，在视觉故事生成方面进行了开创性探索，有望激发更多架构改进方面的研究。

**关键词：**文生图；视频生成；视觉故事生成；主体一致性；

## 1 引言

扩散模型在内容生成领域发展迅速，在图像、3D 对象和视频生成等方面展现出巨大潜力，其生成质量优于之前基于生成对抗网络（GAN）的方法。然而，对于现有模型来说，生成主体一致（如人物身份和服装一致）的图像和视频来描述故事仍然具有挑战性。常用的 IP - Adapter 虽能参考图像引导扩散过程，但文本提示对生成内容的可控性降低；而像 InstantID 和 PhotoMaker 等身份保留方法虽注重身份可控性，但无法保证服装和场景的一致性。

本文旨在找到一种方法，在通过文本提示最大化用户可控性的同时，生成人物身份和服装均一致的图像和视频用于讲故事。为实现这一目标，考虑到使用时间模块虽可保持图像间一致性，但需要大量计算资源和数据，因此本文探索一种轻量级方法，甚至希望能以零样本方式实现。受前人工作启发，发现自注意力对生成视觉内容的整体结构建模至关重要，若能用参考图像引导自注意力计算，有望显著提高图像间一致性，且由于自注意力权重依赖输入，可能无需模型训练或微调，从而提出了一致自注意力（Consistent Self - Attention）方法。

一致自注意力作为 StoryDiffusion 的核心，可零样本插入扩散骨干网络替换原始自注意力，成功保留了生成图像在身份和服装上的一致性，这对讲故事至关重要。对于给定故事文本，先将其划分为多个提示，每个提示对应一张图像，该方法可生成高度一致的图像来讲述故事。为支持长故事生成，结合滑动窗口技术。此外，为将生成的故事帧转换为视频，提出语义运动预测器（Semantic Motion Predictor），它能在语义空间预测两图像间的过渡，生成的视频帧比现有方法更稳定、质量更高。StoryDiffusion 框架结合了这两个组件，能够基于预定义的文本故事生成一致的图像或视频序列，在视觉故事生成方面进行了开创性探索。

## 2 相关工作

在生成模型领域，扩散模型发展迅速且应用广泛，本文主要涉及的文本到图像生成和视频生成方向上也有诸多研究进展，以下将详细介绍相关工作。

### 2.1 扩散模型

扩散模型在近年来的发展势头迅猛，其在生成逼真图像方面的卓越能力已得到充分验证，这使得它在生成建模领域占据主导地位。早期的扩散模型研究主要集中于无条件图像生成，为该领域奠定了理论基础，如 Ho 等人 [1]、Song 等人 [4] 以及 Sohl-Dickstein 等人 [6] 的工作。随着研究的深入，人们不断探索提升扩散模型的效率与性能，在高效采样方法、隐空间去噪、可控性增强以及扩散骨干网络优化等方面持续发力。例如，Song 等人 [5]、Zhang 和 Chen [7] 以及 Lu 等人 [8] 在高效采样方法上取得了显著成果，Rombach 等人 [9] 则在隐空间去噪方面有所建树。与此同时，扩散模型的应用范围也在逐步拓展，在图像生成、视频生成、3D 生成、图像分割以及低层次视觉任务等众多领域都展现出强大的实力。

### 2.2 传统文本到图像生成

文本到图像生成作为扩散模型应用的关键子领域，近期备受瞩目，以 Latent Diffusion [9]、DiT [10] 和 Stable Diffusion XL [11] 等为代表的模型吸引了广泛关注。为增强文本到图像生成的可控性，众多方法应运而生。其中，ControlNet [12] 和 T2I-Adapter 是通过引入深度图、姿态图像或草图等控制条件来指导图像生成的代表性方法；MaskDiffusion [13] 和 StructureDiffusion [14] 则着重提升文本可控性；还有一些工作致力于控制生成图像的布局。根据是否需要测试时微调，这些工作可分为两类：一类仅需用给定图像微调模型的一部分，如 Textual Inversion [15]、DreamBooth [16] 和 Custom Diffusion [17]；另一类则利用在大数据集上预训练的模型，直接使用给定图像控制图像生成。本文的研究重点有所不同，聚焦于保持多图像主体一致性以讲述故事，提出的一致自注意力方法具有训练免费和可插拔的特性，能够在一批图像内建立连接，生成多个主体一致的图像。

### 2.3 传统视频生成

在视频生成领域，鉴于扩散模型在图像生成方面的巨大成功，对视频生成的探索也日益升温。由于文本是用户最直观的描述方式，基于文本的视频生成吸引了最多的关注。VDM [2] 率先将 2D U-Net 从图像扩散模型扩展到 3D U-Net 以实现视频生成，随后 MagicVideo [18] 和 Mindscope [19] 等工作通过引入时间注意力机制，在隐扩散模型基础上减少计算量。Imagen Video [3] 采用级联采样管道。除了传统的端到端文本到视频生成，利用其他条件的视频生成方法也是重要方向，如利用深度图、姿态图、RGB 图像或其他引导运动视频等辅助控制来生成视频，这增强了视频生成的可控性。本文的视频生成方法侧重于过渡视频生成，针对现有过渡视频生成方法在处理复杂过渡（如人物大规模运动）时表现不佳的问题，通过在图像语义空间进行预测，能够更好地处理大运动，从而提升生成视频的质量。

# 3 本文方法

## 3.1 本文方法概述

StoryDiffusion 方法分为两个阶段。第一阶段利用一致自注意力（Consistent Self - Attention）以零样本方式生成主体一致的图像，这些图像可直接用于讲故事或作为第二阶段的输入。第二阶段基于这些一致图像，通过语义运动预测器（Semantic Motion Predictor）创建一致的过渡视频。

在生成主体一致图像时，将一致自注意力插入现有图像生成模型的 U - Net 架构中原自注意力位置，重用原始权重以保持训练免费和可插拔。通过从一批图像中的其他图像采样 token 并与原图像 token 配对，在计算自注意力时跨图像建立联系，使模型在生成过程中促使人物特征收敛，从而生成一致图像。

对于视频生成，语义运动预测器先将条件图像编码到图像语义空间以捕获空间信息，使用预训练的 CLIP 图像编码器将起始帧和结束帧压缩为语义空间向量，再用基于 transformer 的结构预测器进行中间帧预测，最后将预测的语义空间向量作为控制信号，通过视频扩散模型解码为过渡视频。通过在语义空间操作，能更好地建模运动信息，生成高质量过渡视频。

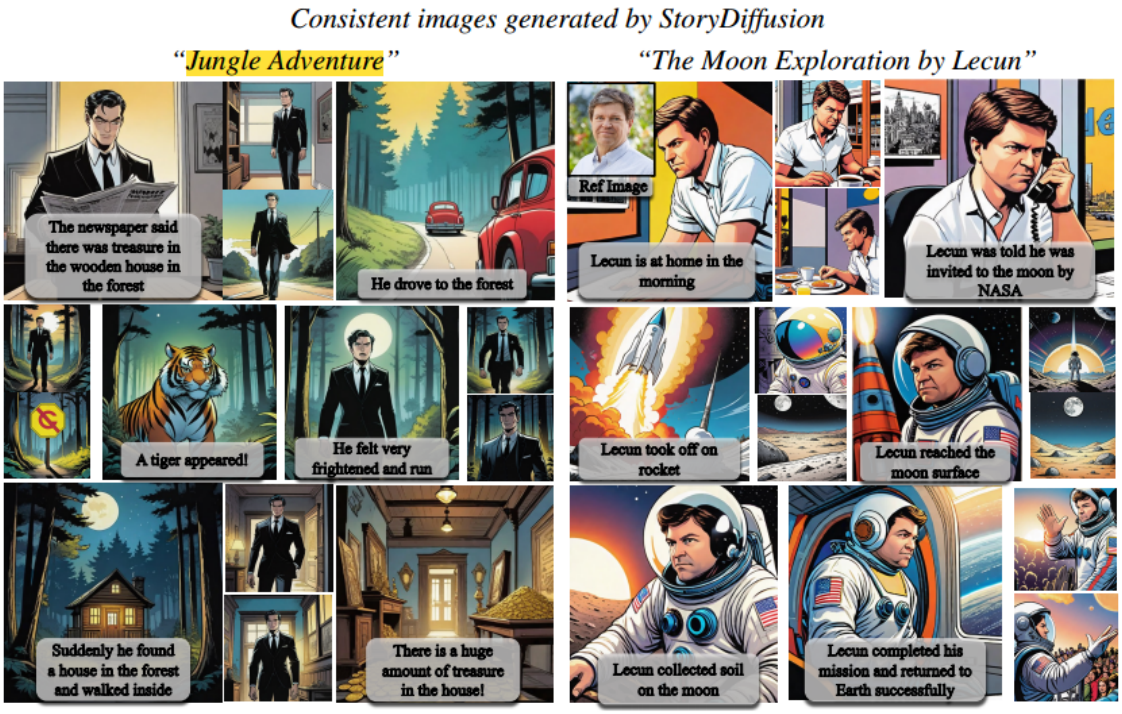


图 1. 生成故事展示

## 3.2 一致性自注意力模块

以无训练的方式生成具有主题一致性的图像，通过在一批图像内建立连接来确保角色一致性。

- **位置插入**：将 Consistent Self-Attention 插入到现有图像生成模型 U-Net 架构中原始自注意力的位置，并重用原始自注意力权重，保持无训练和即插即用的特性。

- **计算过程**：给定一批图像特征 $I \in \mathbb{R}^{B\times N\times C}$（其中 $B$ 为批大小，$N$ 为每个图像中的标记数量，$C$ 为通道数），定义函数 $\text{Attention}(X_k, X_q, X_v)$ 计算自注意力，$X_k$、$X_q$、$X_v$ 分别为注意力计算中的键（Key）、查询（Query）和值（Value）。

原始自注意力在 $I$ 中的每个图像特征 $I_i$ 内独立执行，$I_i$ 被投影到 $Q_i$、$K_i$、$V_i$ 并送入注意力函数，得到

$$O_i = \text{Attention}\left(Q_i, K_i, V_i\right).$$

为建立批内图像间的交互以保持主题一致性，Consistent Self-Attention 从批内其他图像特征中采样一些标记 $S_i$：

$$S_i = \text{RandSample}\left(I_1, I_2, \ldots, I_{i-1}, I_{i+1}, \ldots, I_B\right),$$

其中 RandSample 为随机采样函数。

采样后，将采样标记 $S_i$ 与图像特征 $I_i$ 配对形成新的标记集 $P_i$，对 $P_i$ 进行线性投影以生成 Consistent Self-Attention 的新键 $K_{P_i}$ 和值 $V_{P_i}$，原始查询 $Q_i$ 不变，最后计算自注意力：

$$O_i = \text{Attention}\left(Q_i, K_{P_i}, V_{P_i}\right).$$

尽管方式简单且无需训练，但能有效生成主题一致的图像，促进模型在生成过程中对角色、面部和服装的一致性收敛，使生成的图像能更好地用于讲述复杂故事。

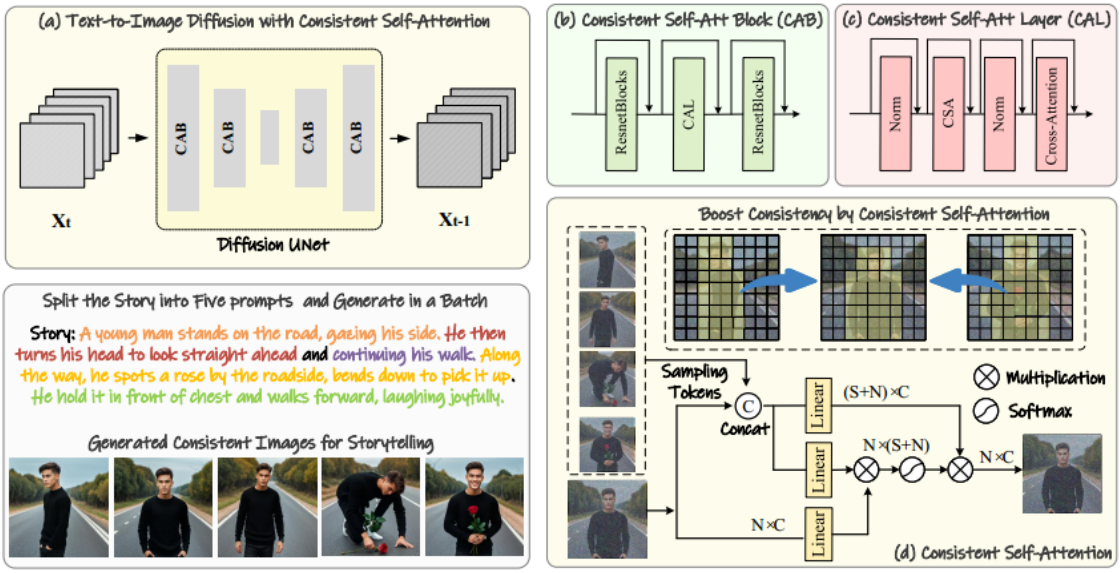


图 2. 自注意力模块示意图

### 3.3 空间运动预测模块

将生成的具有角色一致性的图像序列转换为具有平滑过渡和一致主题的视频，解决现有方法在处理大动作过渡时的不足。

- **图像编码到语义空间**：利用预训练的 CLIP 图像编码器 $E$ 将给定的起始帧 $F_s$ 和结束帧 $F_e$ 压缩为图像语义空间向量 $K_s$、$K_e$，即

$$K_s, K_e = E\left(F_s, F_e\right).$$

- **语义空间运动预测**：在图像语义空间中，使用基于 Transformer 的结构预测器对每个中间帧进行预测。首先通过线性插值将 $K_s$ 和 $K_e$ 扩展为序列 $K_1, K_2, \cdots, K_L$（$L$ 为所需视频长度），然后将该序列送入一系列 Transformer 块 $B$ 中预测过渡帧：

$$P_1, P_2, \cdots, P_L = B\left(K_1, K_2, \cdots, K_L\right).$$

- **解码为过渡视频**：将预测的图像语义嵌入 $P_1, P_2, \cdots, P_L$ 作为控制信号，视频扩散模型作为解码器。在扩散过程中，对于每个视频帧特征 $V_i$，将文本嵌入 $T$ 和预测的图像语义嵌入 $P_i$ 连接起来，计算交叉注意力：

$$V_i = \text{CrossAttention}\left(V_i, \text{concat}\left(T, P_i\right), \text{concat}\left(T, P_i\right)\right).$$

- **模型优化**：通过计算 $L$ 帧预测过渡视频 $O = (O_1, O_2, \cdots, O_L)$ 和 $L$ 帧真实值 $G = (G_1, G_2, \cdots, G_L)$ 之间的均方误差（MSE）损失来优化模型：

$$\text{Loss} = \text{MSE}(G, O).$$

通过将图像编码到语义空间来整合空间位置关系，从而更好地建模运动信息，能够生成具有大动作的平滑过渡视频，在实验中表现出显著优于现有方法的性能。

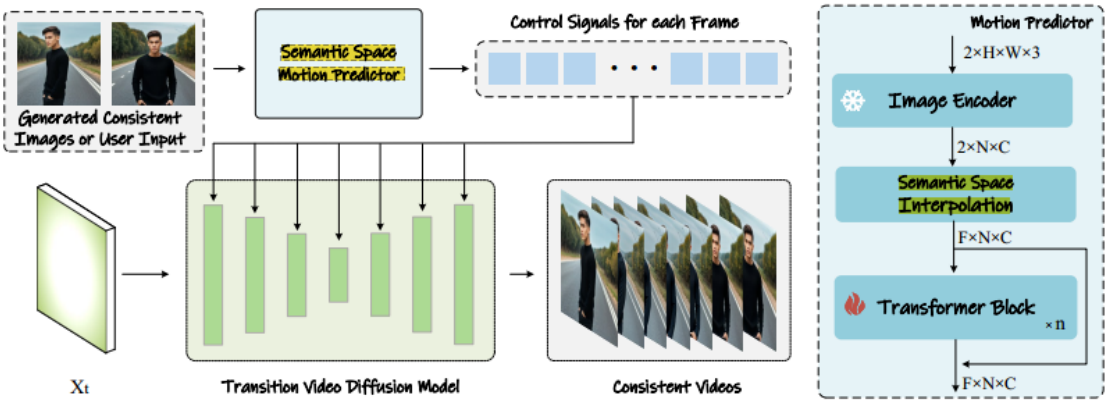


图 3. 运动预测模块示意图

## 4 复现细节

### 4.1 与已有开源代码对比

StoryDiffusion 虽然在生成故事方面表现良好，尤其是生成具有主体一致性和视觉连贯性的图像序列，但在情感故事生成领域存在以下不足之处：1. 对于多样化情感表达的支持较弱，生成的故事在情感传递上略显单调。2. 在生成长时间序列的视频或图像时，过渡不够平滑，导致某些场景的大动作或复杂情节表现欠佳。3. 对用户输入依赖较高，需要用户手动定制完整的故事流程和细节，不够智能化。

由于 StoryDiffusion 有完善的官方开源代码，本次复现过程中约 70% 的工作量直接引用了源代码，主要用于维持故事生成所需的主体一致性和基础功能。此外，本次复现工作在已有代码基础上，结合实际需求进行了以下显著优化与改进。

### 4.2 创新点

为了弥补上述不足，复现工作主要在以下两个方面进行了创新与改进：

1. **同批次共享注意力掩码改进：**在原始代码中，掩码生成采用了随机全组采样的方式，可能导致图像生成在连续帧之间的过渡性不足，影响故事的渐进性和连贯性。为了解决这一问题，本工作根据时间进度对掩码生成策略进行了改进：- 在生成掩码时，优先根据时间进度采样前几帧图像，强化了帧与帧之间的上下文关联性和故事逻辑连续性。- 引入优化的随机值分布，提高采样的稳定性和生成的主体一致性。这一改进使得故事情节的过渡更加自然流畅，同时显著提升了生成故事画面的整体质量和一致性。

```python
def cal_attn_mask_xl(total_length,id_length,sa32,sa64,height,width,device="cuda",dtype= torch.float16):
    nums_1024 = (height // 32) * (width // 32)  # 每层 patch 为 32 * 32， 有 576 个 patchs / tokens
    nums_4096 = (height // 16) * (width // 16)  # 2304 个

    bool_matrix1024 = torch.rand((1, total_length * nums_1024),device = device,dtype = dtype) < sa32
    bool_matrix4096 = torch.rand((1, total_length * nums_4096),device = device,dtype = dtype) < sa64
    bool_matrix1024 = bool_matrix1024.repeat(total_length,1)
    bool_matrix4096 = bool_matrix4096.repeat(total_length,1)  # [total_length, total_length * nums_1024]

    for i in range(total_length):
        bool_matrix1024[i:i+1,id_length*nums_1024:] = False   # 设置时间步，不关注 id_length = 4 以后的区域
        bool_matrix4096[i:i+1,id_length*nums_4096:] = False
        bool_matrix1024[i:i+1,i*nums_1024:(i+1)*nums_1024] = True  # 自关注保证全关注全为 True
        bool_matrix4096[i:i+1,i*nums_4096:(i+1)*nums_4096] = True

    #save_mask_as_image(bool_matrix1024.unsqueeze(1).reshape(total_length, total_length, nums_1024), 1024,
     (width // 32)))
    #save_mask_as_image(bool_matrix4096.unsqueeze(1).reshape(total_length, total_length, nums_4096), 4096,
     (width // 16)))

    mask1024 = bool_matrix1024.unsqueeze(1).repeat(1,nums_1024,1).reshape(-1,total_length * nums_1024)
    mask4096 = bool_matrix4096.unsqueeze(1).repeat(1,nums_4096,1).reshape(-1,total_length * nums_4096)
    # [total_length * nums_1024, total_length * nums_1024]

    # 注意力掩码形状=[Batch_Size,Query_Tokens,Key_Tokens]=[4,1024,1024], 通常 Query_Tokens=Key_Tokens
    # 多头则 [Batch_Size, head, Query_Tokens,Key_Tokens]
    # 如果涉及跨 prompt 的注意力机制，掩码的形状会进一步扩展到考虑 整个批量的 token 总数
```

图 4. 改进后的注意力掩码示意图

2. **引入大语言模型进行情感故事 prompt 生成：**针对原始模型过度依赖用户手动定制详细故事流程的问题，本次工作引入了预训练的大语言模型（如 GPT 系列模型）作为情感故事生成工具。主要实现方式如下：- 通过用户输入的单个情感词（例如“悲伤”、“欢快”），生成一段具有明确情感基调的故事文本 prompt。- 使用生成的文本 prompt 引导图像生成过程，从而实现更具情感表达和故事多样性的画面序列生成。- 提升了用户体验，降低了生成高质量情感故事的门槛。

这一改进不仅增强了生成的故事对情感基调的表达能力，也极大丰富了故事的多样性和创意性，尤其在开放式情感生成任务中表现出色。

```python
theme = "StoryWeaver"
elements = "cookies, sweeps the floor, amusement park ride, roller coaster, christmas ornaments, night sky"
general_prompt = "A white bear"
prompt_array = [
    f"{general_prompt} busily prepares a batch of cookies in the kitchen",
    f"{general_prompt} diligently sweeps the floor of his cozy wooden house",
    f"{general_prompt} engage in a game of soccer, relishing every moment of the thrilling match",
    f"{general_prompt} engage in an intense game of tennis, his skills on full display",
    f"{general_prompt} skillfully prepares refreshing fruit juices, his face beaming with satisfaction",
    f"{general_prompt} sits on a wooden bench, captivated by the joy around him"
]
negative_prompt = "deformed, bad anatomy, disfigured, poorly drawn face, mutation, extra limb, ugly, disgusting, poorly drawn hands,
 missing limb, floating limbs, disconnected limbs, blurry, watermarks, oversaturated, distorted hands, amputation"

### Set the generated Style
style_name = "Japanese Anime" #定义风格
# (No style)
# Japanese Anime
# Comic book

def apply_style_positive(style_name: str, positive: str):
    p, n = styles.get(style_name, styles[DEFAULT_STYLE_NAME])
    return p.replace("{prompt}", positive)
def apply_style(style_name: str, positives: list, negative: str = ""):
    p, n = styles.get(style_name, styles[DEFAULT_STYLE_NAME]) # 后者是默认风格
    return [p.replace("{prompt}", positive) for positive in positives], n + ',' + negative  # 加入消极抑制 prompt
```

图 5. 情感 prompt 工程示意图

### 4.3 总结

通过以上两点改进，本次复现工作不仅优化了现有方法在故事生成中的连贯性和一致性，还为情感表达和开放式生成提供了新的可能性。这些创新为模型的实际应用场景拓展了更多空间，显著提升了生成质量和用户交互体验。

## 实验结果对比

通过对比 Emo-Score 和 Image-Text Similarity 两项指标，可以清晰地看出本模型的改进效果：

1. **情感得分（Emo-Score）**
   Emo-Score 反映了故事生成的情感丰富性和情感表达的深度。在引入大语言模型进行情感激活后，情感得分显著提升，这表明模型能够更好地捕捉和表达故事中的情感元素，为生成的故事增添了更强的感染力和情感表现力。

2. **图文相似度（Image-Text Similarity）**
   Image-Text Similarity 评估生成图像与文本描述之间的匹配度，体现了故事生成的多样性和表达准确性。通过改进同批次共享注意力掩码的机制，模型在图文表达的相关性上取得了一定的提升。这种改进使得生成的图像能够更准确地反映文本描述，同时增强了生成故事的连贯性与一致性。

3. **效果总结**
   实验结果表明，引入的大语言模型在激发情感故事生成方面效果显著，而注意力掩码的优化进一步提升了模型对故事多样性和细节表达的能力。综合来看，这些改进充分证明了所提出方法的有效性，为情感故事生成任务提供了更强的支持。

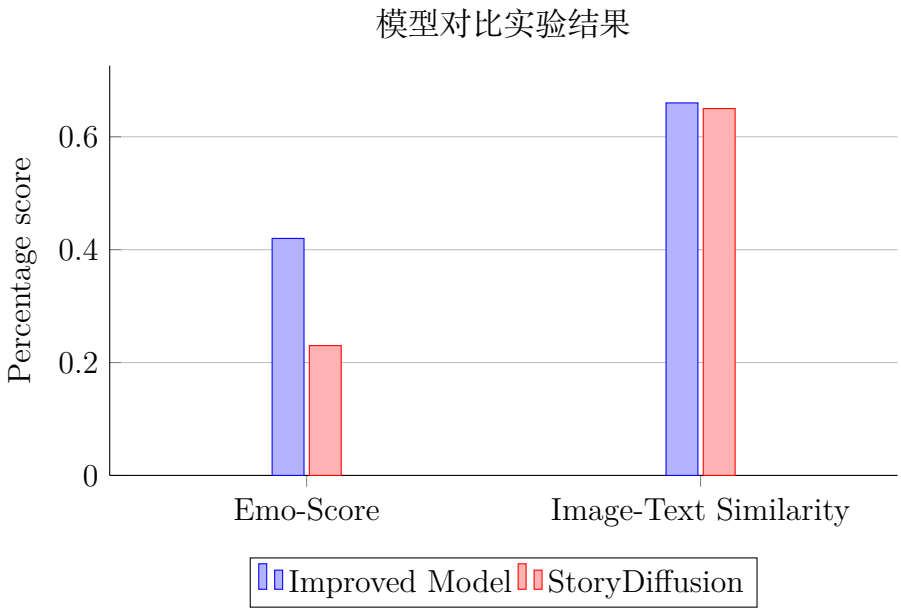


图 6. 模型对比

## 5 总结与展望

本次复现工作优化了现有方法在故事生成中的连贯性和一致性，同时引入了情感故事生成的新机制，为情感表达和开放式生成提供了新的可能性。然而，改进模型的创新性和整体工作量仍有提升空间。

目前的改进尚未充分解决大语言模型在情感理解上的局限性，生成的故事在情感表达的细腻程度和多样性方面仍有提升余地。此外，在故事生成过程中，主角的一致性未能始终稳定保持，个别帧之间仍存在细微的不一致性，这也成为后续优化的重要方向。

未来的工作将继续聚焦于情感故事生成这一领域，致力于进一步改进模型性能。一方面，增强模型对情感表达的理解和表现能力，使生成的故事更具情感深度和感染力；另一方面，优化主体一致性相关机制，使主角在整个生成过程中始终保持高度一致。同时，还将进一步提升故事的逻辑性与表达流畅性，为复杂情感和叙事需求提供更加完善的解决方案。

## 参考文献

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[2] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint*, 2022a.

[3] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint*, 2022b.

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint*, 2020.

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[6] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[7] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *ICLR*, 2023.

[8] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[10] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint*, 2022.

[11] 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. *arXiv preprint*, 2023.

[12] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[13] Yupeng Zhou, Daquan Zhou, Zuo-Liang Zhu, Yaxing Wang, Qibin Hou, and Jiashi Feng. Maskdiffusion: Boosting text-to-image consistency with conditional mask. *arXiv preprint*, 2023.

[14] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *ICLR*, 2023.

[15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint*, 2022.

[16] 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 *CVPR*, 2023.

[17] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023.

[18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint*, 2023.

[19] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. *arXiv preprint*, 2023.