StoryGPT-V: Large Language Models as Consistent Story Visualizers
– Supplementary Materials –

The supplementary material provides:

- Section 1: Multi-modal story generation ability.
- Section 2: First stage ablation; Number of `[IMG]` tokens; Different LLMs.
- Section 3: The limitation of CLIP score; Human Evaluation; Open Domain Evaluation on VIST.
- Section 4: Human evaluation.
- Section 5: Data preparation and implementation details.
- Section 6: Limitation discussion.
- Section 7: Responses to Rebuttal.
- Section 8: Qualitative results.

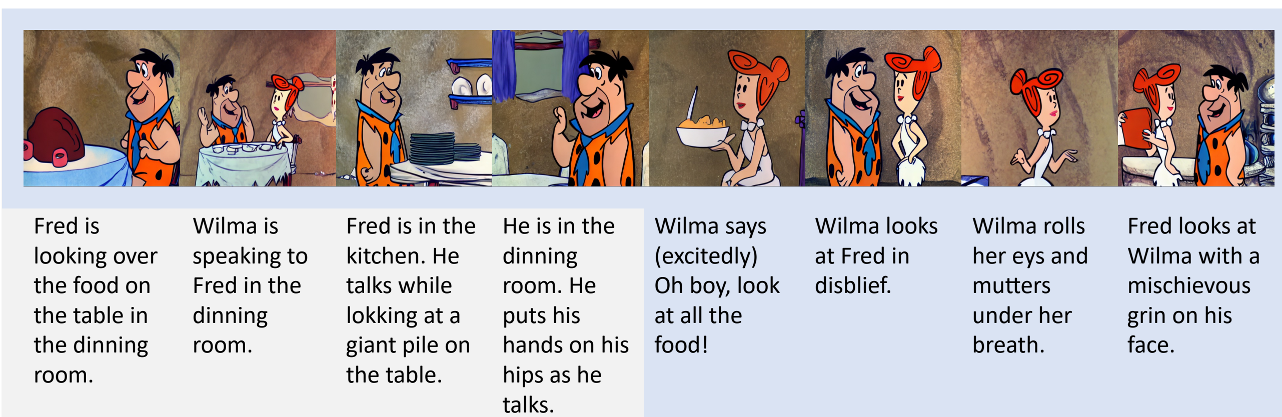

Figure 1. **Our model StoryGPT-V extending stories in both language and vision:** Gray part is the text descriptions from datasets. Blue part corresponds to the model-generated frames and the continued written stories based on the previous captions.

1. Multi-modal Story Generation

Owing to StoryGPT-V design leveraging the advanced capabilities of Large Language Models (LLMs), it exhibits a unique proficiency in that it can extend visual stories. StoryGPT-V is not merely limited to visualizing stories based on provided textual descriptions. Unlike existing models, it also possesses the innovative capacity to extend these narratives through continuous text generation. Concurrently, it progressively synthesizes images that align with the newly generated text segments.

Figure 1 presents an example of a multi-modal story generation. Initially, the first four frames are created according to the text descriptions from the FlintstonesSV [1] dataset (gray part). Subsequently, the model proceeds to write the description for the next frame (blue part), taking into account the captions provided earlier, and then creates a frame based on this new description (blue part). This method is employed iteratively to generate successive text descriptions and their corresponding frames.

Our model represents a notable advancement in story visualization, being the first of its kind to consistently produce both high-quality images and coherent narrative descriptions. This innovation opens avenues for AI-assisted technologies to accelerate visual storytelling creation experiences by exploring various visualized plot extensions as the story builds.

2. Ablation Studies

2.1. Effect of first-stage design.

In Table 1 lower half, we conducted an ablation study on how the stage-1 design contributes to the final performance. In the first line, the stage-2 LLM is aligned with vanilla LDM fine-tuned on FlintstonesSV [1]. The second line aligns the LLM output with our Char-LDM's text embedding (Emb$_{text}$), while the last line aligns with character-augmented fused embedding (Emb$_{fuse}$) of our Char-LDM. The first two lines align to the same text embedding encoded by the CLIP [12] text encoder, however, our Char-LDM enhanced with cross-attention control ($\mathcal{L}_{reg}$) produces more precise characters. Different from Emb$_{text}$, the last line is aligned with Emb$_{fuse}$, which is augmented with characters' visual features. This visual guidance helps LLM to interpret references more effectively by linking "he, she, they" to the previous language and image context.

Models	Aligning space	Char-Acc (↑)	Char-F1 (↑)	BG-Acc (↑)	BG-F1 (↑)	FID (↓)
Vanilla LDM [14]	×	75.37	87.54	52.57	58.41	32.36
Our Stage-2	Vanilla LDM Emb$_{text}$	84.06	92.54	53.18	58.29	22.94
	Char-LDM Emb$_{text}$	86.10	93.46	54.92	60.15	21.30
	Char-LDM Emb$_{fuse}$ (default)	88.45	94.94	56.45	62.09	21.71

Table 1. The output of our stage-2 model (OPT) is aligned with conditional input of vanilla LDM [14] (finetuned on FlintstonesSV [1]), our Char-LDM text embedding (Emb$_{text}$) or character-augmented fused embedding (Emb$_{fuse}$).

2.2. Number of `[IMG]` Tokens

We further examined the impact of the number of added `[IMG]` tokens. As indicated in Table 2, aligning with the fused embedding and setting $R = 8$ yields the best performance.

Models	R	Char-Acc (↑)	Char-F1 (↑)	BG-Acc (↑)	BG-F1 (↑)	FID (↓)
Emb$_{text}$	4	82.14	90.18	54.28	59.58	21.33
Emb$_{text}$	8	86.10	93.46	54.92	60.15	21.30
Emb$_{text}$	16	83.77	91.07	54.08	60.21	21.58
Emb$_{fuse}$	4	86.23	93.43	54.57	59.61	21.97
Emb$_{fuse}$	8	88.45	94.94	56.45	62.09	21.71
Emb$_{fuse}$	16	85.35	91.96	52.93	58.86	23.73

Table 2. StoryGPT-V Ablations: Impact of R, the number of added `[IMG]` tokens. Emb$_{text}$: the output of LLM (OPT) is aligned with text embedding extracted from the text encoder; Emb$_{fuse}$: aligned with fused embedding Emb$_{fuse}$ of first stage model.

2.3. Different LLMs (OPT vs Llama2)

Models	# Params	Char-Acc (↑)	Char-F1 (↑)	BG-Acc (↑)	BG-F1 (↑)	FID (↓)	BLEU4 (↑)	CIDEr (↑)
OPT [17]	6.7b	88.45	94.94	56.45	62.09	21.71	0.5037	1.6718
Llama2 [15]	7b	89.08	95.07	57.29	62.62	21.56	0.5169	1.7516

Table 3. Performance on FlintstonesSV [1] dataset with referential text using different LLMs.

Our primary contribution lies in leveraging Large Language Models (LLMs) for reference resolution for consistent story visualization. In our work, we experimented with OPT-6.7b[1] and Llama2-7b-chat[2] models. It's important to note that the

[1] https://huggingface.co/facebook/opt-6.7b
[2] https://huggingface.co/meta-llama/Llama-2-7b-chat

utilization of Llama2 was specifically to demonstrate its additional capability for multi-modal generation. The ablation study of different LLMs was not the main focus of our research.

Our findings, as illustrated in Table 3, indicate only a slight improvement when changing from OPT [17] to Llama2 [15]. This marginal difference is attributed to the evaluation metric's emphasis on image-generation capabilities, which assesses whether the model's visual output aligns well with first-stage Char-LDM's conditional input space.

3. Evaluation

3.1. Text-image alignment.

CLIP [12] is trained on large-scale image-caption pairs to align visual and semantic space. However, a domain gap exists between pre-train data and the story visualization benchmark. Therefore, we finetune CLIP [12] on the story visualization datasets. However, we found it still hard to capture fine-grained semantics, either text-image (T-I) similarity or image-image similarity (I-I), i.e., the similarity between visual features of generated images and corresponding ground truth images.

Upon this observation, we choose the powerful captioning model BLIP2 [6] as the evaluation model. We finetune BLIP2 on FlintstonesSV [1] and PororoSV [7], respectively, and employ it as an image captioner for generated visual stories. We avoided direct comparisons to bridge the gap between BLIP2's predictions and the actual ground truth captions. Instead, we used the fine-tuned BLIP2 to generate five captions for each ground truth image and one caption for each generated image. and report average BLEU4 [10] or CIDEr [16] score based on these comparisons.

Models	CLIP (T-I) (↑)	CLIP (I-I) (↑)	BLEU4 (↑)	CIDEr (↑)
StoryDALL-E [8]	0.4417	0.8112	0.4460	1.3373
LDM [14]	0.5007	0.8786	0.4911	1.5103
Story-LDM [13]	0.4979	0.8795	0.4585	1.4004
StoryGPT-V (Ours OPT)	0.5106	0.889	0.5070	1.6607

Table 4. Text-image alignment score for FlintstonesSV [1] with referential text descriptions in terms of CLIP [12] similarity, BLEU4 [10] and CIDEr [16].

4. Human evaluation.

we use Mechanical Turk to assess the quality of 100 stories produced by our methods or Story-LDM [13] on FlintStonesSV [1]. Given a pair of stories generated by Story-LDM [13] and our model, MTurkers are asked to decide which generated four-frame story is better w.r.t visual quality, text-image alignment, character accuracy, and temporal consistency. Each pair is evaluated by 3 unique workers. In Figure 2, our model demonstrates significantly better story visualization quality with accurate and temporally coherent synthesis. The human study interface is illustrated in Figure 3.

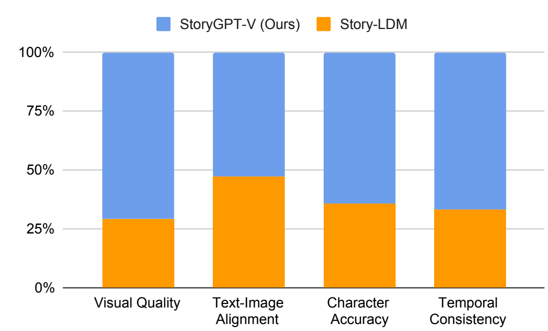

Figure 2. Human evaluation results on FlintStonesSV [1] w.r.t visual quality, text-image alignment, character accuracy and temporal consistency.

4.1. Open Domain Evaluation

We mainly focus on closed-domain story visualization and character synthesis with ambiguous references. VIST is a story visualization data but lacks consistent visual stories as it relies on people crafting stories for 5 selected photos from a Flickr album. And it doesn't contain character/background labels for a comprehensive evaluation in the setting of consistent story visualization like [1]. We report CLIP image similarity and LPIPS score following [5] in Table 5.

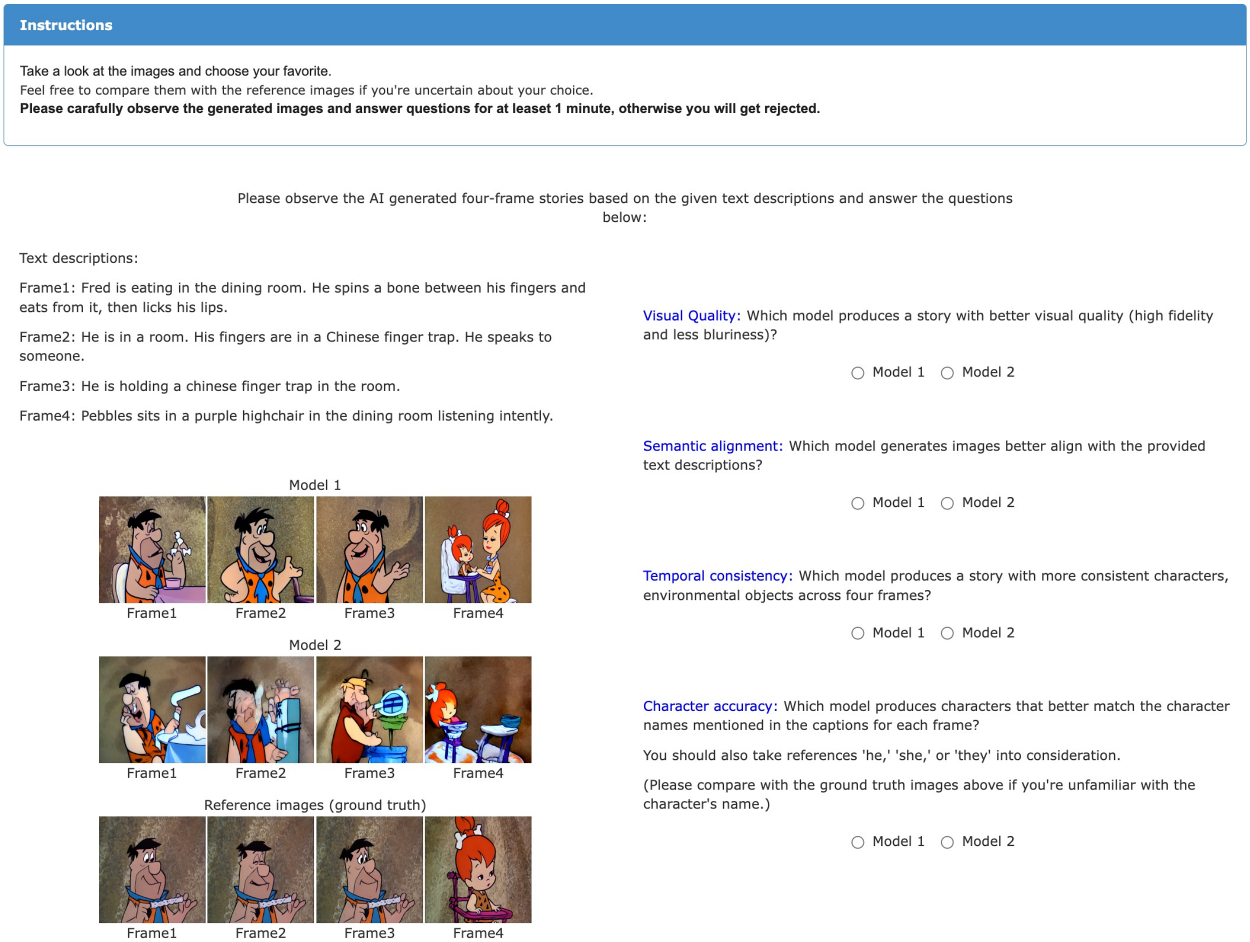

Figure 3. Human study interface.

5. Implementation Details

5.1. Data preparation

FlintstonesSV [1] provides the bounding box location of each character in the image. We fed the bounding boxes into SAM [4] to obtain the segmentation map of corresponding characters. This offline supervision from SAM is efficiently obtained without the need for manual labeling efforts.

Models	CLIP-I (↑)	LPIPS (↓)
LDM [14]	0.598	0.704
Story-LDM [13]	0.504	0.715
StoryGPT-V (Ours)	0.613	0.692

Table 5. Results on VIST [3] dataset.

5.2. Extending dataset with referential text

We follow Story-LDM [13] to extend the datasets with referential text by replacing the character names with references, i.e., he, she, or they, wherever applicable as shown in Algorithm 1. The statistics before and after the referentail extension are shown in Table 6. Please refer to Story-LDM [13] implementation[3] for more details on how the referential dataset is extended.

5.3. First stage training

We built upon pre-trained Stable Diffusion [14] v1-5[4] and use CLIP [11] ViT-L to extract characters' visual features. We freeze the CLIP text encoder and fine-tune the remaining modules for 25,000 steps with a learning rate of 1e-5 and batch size

[3]https://github.com/ubc-vision/Make-A-Story/blob/main/ldm/data
[4]https://huggingface.co/runwayml/stable-diffusion-v1-5

Dataset	# Ref (avg.)	# Chars	# Backgrounds
FlintstonesSV [1]	3.58	7	323
Extended FlintstonesSV	4.61	7	323
PororoSV [7]	1.01	9	None
Extended PororoSV	1.16	9	None

Table 6. Dataset statistics of FlintstonesSV [1] and PororoSV [7]

of 32. The first stage utilizes solely the original text description without extended referential text. To enhance inference time robustness and flexibility, with or without reference images, we adopt a training strategy that includes 10% unconditional training, i.e., classifier-free guidance [2], 10% text-only training, and 80% augmented text training, which integrates visual features of characters with their corresponding token embeddings.

5.4. Second stage training

We use OPT-6.7B[5] model as the LLM backbone in all experiments in the main paper. To expedite the second stage alignment training, we first pre-compute non-referential fused embeddings residing in the input space of the first-stage Char-LDM. We map visual features into $m = 4$ token embeddings as LLM input, set the max sequence length as 160 and the number of additional `[IMG]` tokens as $R = 8$, batch size as 64 training for 20k steps. Llama2 is only trained for the experiments highlighted in the supplementary materials, demonstrating its capability for multi-modal generation and the ablation of different LLMs. The training configuration is almost the same as OPT, except for batch size 32. All experiments are executed on a single A100 GPU.

Please refer to all the details at the source code.

Algorithm 1 Character Replacement Algorithm

Definitions:
i: index for frames, ranging from 1 to N
S_i: text description of frame i
$\mathcal{C}_i$: a set contains immediate character(s) in the current frame
for $i \in \{1, 2, \ldots, N\}$ **do**
  if $i = 1$ **then**
    $\mathcal{C}_i \leftarrow$ immediate character of S_i
  else
    if $\mathcal{C}_i \subseteq \mathcal{C}_{i-1}$ **then**
      if $\text{length}(\mathcal{C}_i) = 1$ **then**
        Replace $\mathcal{C}_i$ in S_i with "he" or "she"
      else if $\text{length}(c) > 1$ **then**
        Replace $\mathcal{C}_i$ in S_i with "they"
      end if
    end if
    $\mathcal{C}_i \leftarrow \mathcal{C}_{i-1}$
  end if
end for

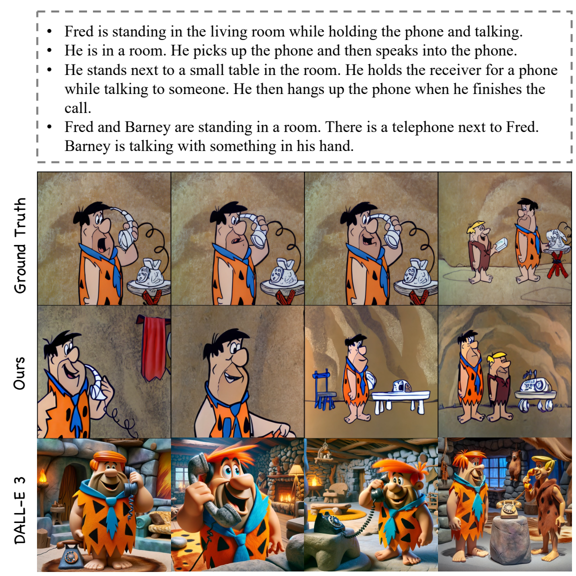

Figure 4. DALL-E 3 [9] zero-shot inference on FlintstonesSV [1] dataset.

6. Limitations

Our method demonstrates proficiency in resolving references and ensuring consistent character and background conditions in the context provided by guiding the output of a multi-modal Large Language Model (LLM) with character-augmented semantic embedding. However, several limitations remain. The process involves feeding the previously generated frame into the LLM to produce a visual output that aligns with the Latent Diffusion Model (LDM) input conditional space. This

[5] https://huggingface.co/facebook/opt-6.7b

approach guarantees semantic consistency, enabling the generation of characters and environmental objects that resemble their originals. Nonetheless, there are minor discrepancies in detail. This is because the visual output from the Large Language Model (LLM) is aligned with the semantic embedding space rather than the pixel space, which hinders the complete reconstruction of all elements in the input image. However, the current most powerful multi-modal LLM, i.e., DALL-E 3 [9], could not solve this exact appearance replication in the multi-round image generation task (Figure 4), indicating an area ripe for further exploration and research.

7. Response to Rebuttal

How does the proposed method maintain background consistency?

We introduce $\mathcal{L}_{img}$ (Eq.7) to provide pixel-level supervision for maintaining visual consistency during the second stage training. Specifically, the visual prediction $\texttt{[IMG}_{1-R}\texttt{]}$ of the current frame generated from LLM, is conditioned on the contextual input from the previous frames. Then Char-LDM utilizes the visual output from the LLM as guidance during the denoising process to generate the current frame. The loss function $\mathcal{L}_{img}$ enforces gradient propagation to the LLM, encouraging $\texttt{[IMG}_{1-R}\texttt{]}$ to preserve contextual consistency and generate frames closely aligned with the ground truth.

How to operate when the character mask is not available during inference?

Character masks are only used during the first stage of training to guide attention in Char-LDM for accurate character generation. In the second stage, our model only takes contextual information (previous frames and captions) and the current caption as input to generate the current frame, without requiring any masks. Therefore, the inference stage operates entirely mask-free.

The structure and training methods of LLM Mapper and LDM Mapper.

We mentioned in line 315 that $\texttt{Mapper}_{LLM}$ is a linear layer with trainable matrix $\mathbf{W}_{v2t}$ mapping from visual feature to LLM's input space commonly used in MLLMs [25,62]. We detailed the structure of $\texttt{Mapper}_{LDM}$ in line 332-337 that it is a 4-layer encoder-decoder Transformer model similar to BLIP-2 QFormer [22]. Both modules are updated during the second-stage training while keeping the LLM frozen. Additionally, we have included an anonymous link to the code implementation in the main paper for reference.

The effectiveness of the LLM's performance.

The LLM significantly enhances coreference resolution in story visualization. While our Char-LDM struggles with ambiguous pronouns (e.g., he, she, they), our StoryGPT-V leverages the LLM's strong reasoning ability to accurately generate stories from ambiguous descriptions as shown below. We also investigate different LLMs in Tab.3 (supp).

Models	Char-Acc (↑)	Char-F1 (↑)	BG-Acc (↑)	BG-F1 (↑)	FID (↓)
Char-LDM (Ours w/o LLM)	83.51	90.45	55.31	61.93	21.96
StoryGPT-V (Ours)	88.45	94.94	56.45	62.09	21.71

The multi-stage architecture (Char-LDM with SAM + LLM) introduces computational demands.

Please kindly note that SAM is only used only to obtain the segmentation masks in data processing, and is not involved in training stage. **Training:** In the first stage, we train our Char-LDM model with 0.5 billion trainable parameters for 32 GPU hours. During the second stage, the LLM remains frozen while the LLM Mapper, LDM Mapper, and the embeddings for the additional tokens $\texttt{[IMG}_{1-R}\texttt{]}$ are updated. This stage involves only 0.2 billion trainable parameters for 24 GPU hours training. The LLM does not introduce much computation overhead. When generating 4 frames, our model takes up additional memory due to LLM, i.e., 15.86GB (Story-LDM) vs 25.05GB (Ours). However when increasing the number of generated frames (e.g., 40), our model achieves faster, more memory-efficient inference with improved accuracy as shown below.

Models	Speed (↓)	GPU-Memory (↓)	Char-Acc (↑)	FID (↓)
Story-LDM	225.75 sec	75.92 GB	63.40	60.33
StoryGPT-V (Ours)	108.54 sec	26.10 GB	81.04	48.37

Performance on Mugen dataset.

MUGEN is not widely used in story visualization task, since it has only 3 characters and 6 backgrounds. We add our results below.

How much diverse the proposed method is?

Models	Char-Acc (↑)	Char-F1 (↑)	BG-Acc (↑)	BG-F1 (↑)	FID (↓)
Story-LDM	93.40	95.60	92.19	92.37	62.16
StoryGPT-V (Ours)	93.92	96.14	93.21	93.80	54.75

Our method leverages pretrained knowledge to generate diverse environmental objects in the story domain.

Comparing with SOTA methods, how many frames are used to generate a coherent story.

During inference, the first frame is generated solely from the first caption, and subsequent frames are autoregressively generated using contextual information (previous generated frames and captions) and the current caption. We evaluate on 4 frames following previous setup, but also extend up to 40.

8. Qualitative Results

We provide more generated samples on FlintstonesSV [1] and PororoSV [7] with referential text as Figure 5-14 show.

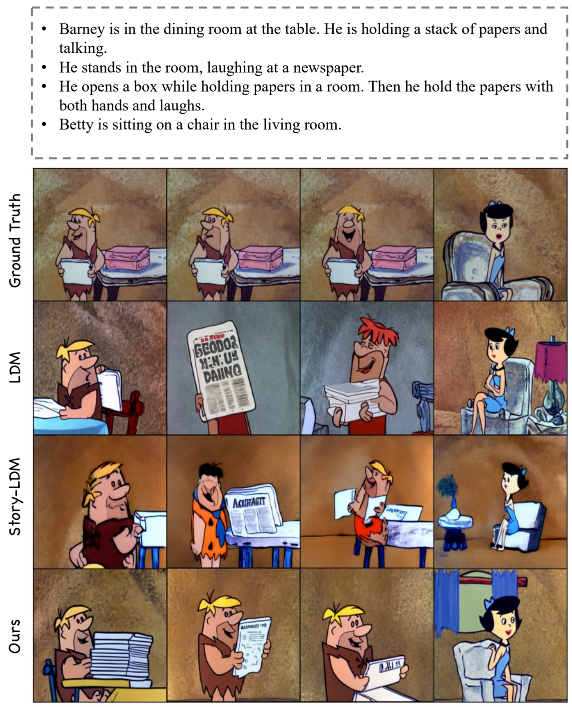

Figure 5. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

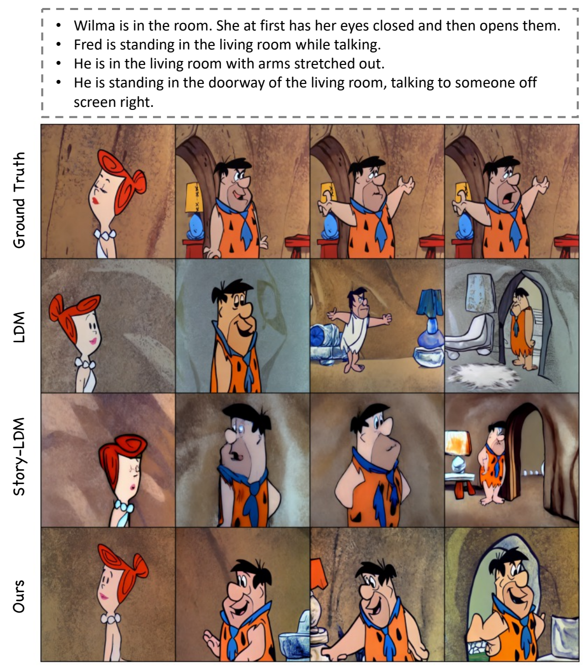

Figure 6. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

References

[1] Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. Imagine this! scripts to compositions to videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 598–613, 2018. 1, 2, 3, 4, 5, 7, 8, 9

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[3] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1233–1239, 2016. 4

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4

[5] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. *arXiv preprint arXiv:2305.17216*, 2023. 3

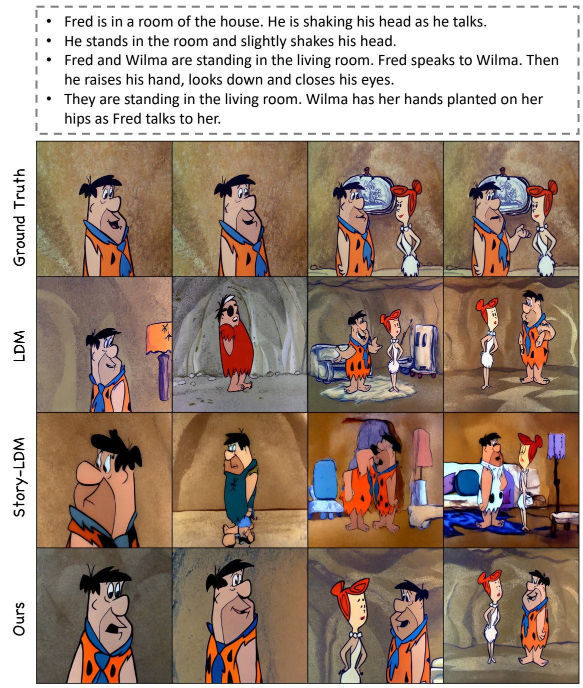

Figure 7. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

Figure 8. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

Figure 9. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

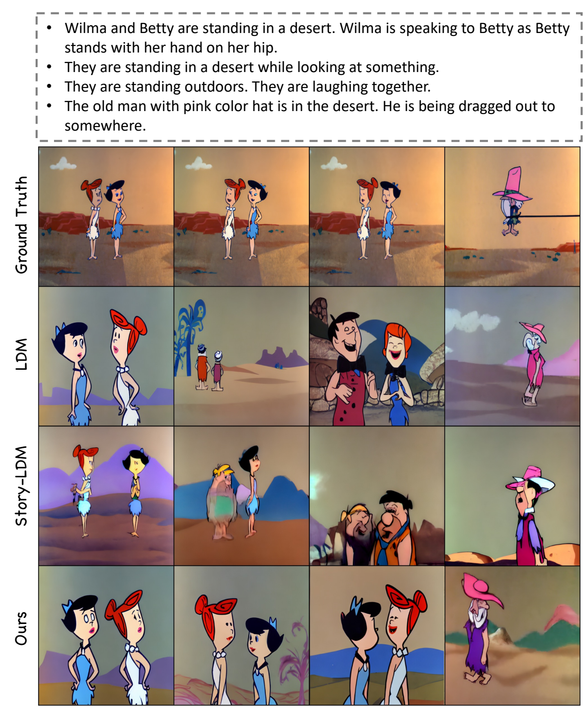

Figure 10. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

[6] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

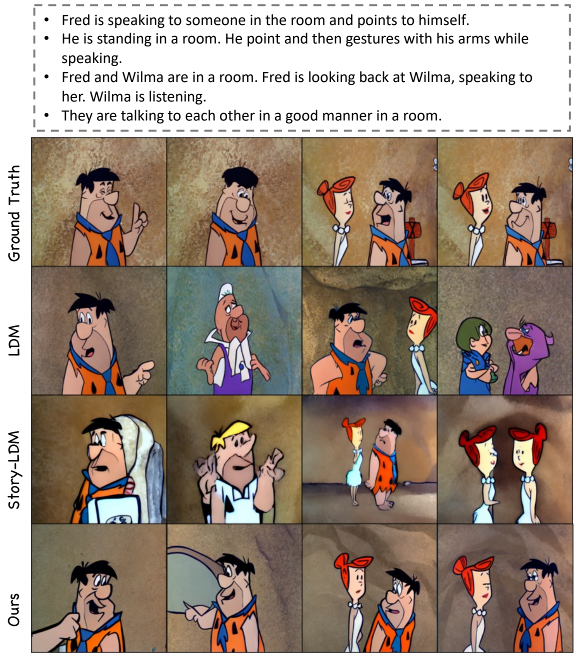

Figure 11. Qualitative comparison on FlintstonesSV [1] with co-reference descriptions.

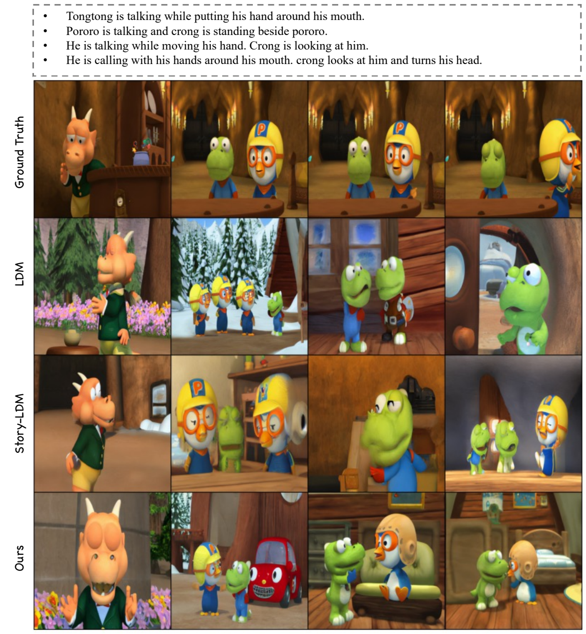

Figure 12. Qualitative comparison on PororoSV [7] with co-reference descriptions.

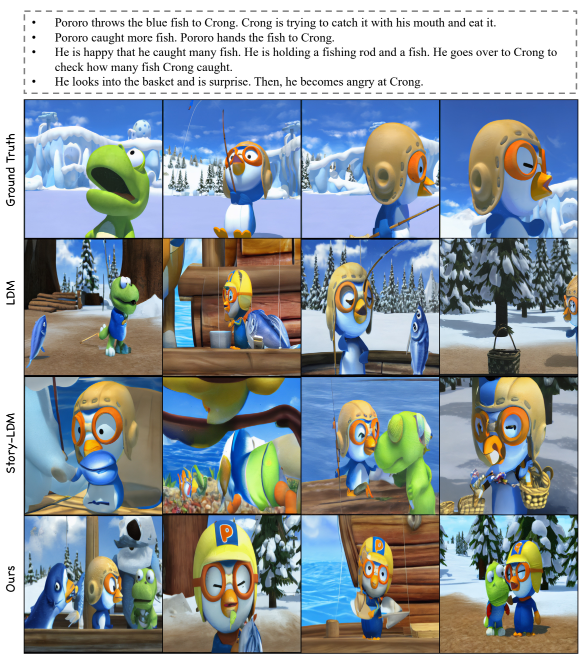

Figure 13. Qualitative comparison on PororoSV [7] with co-reference descriptions.

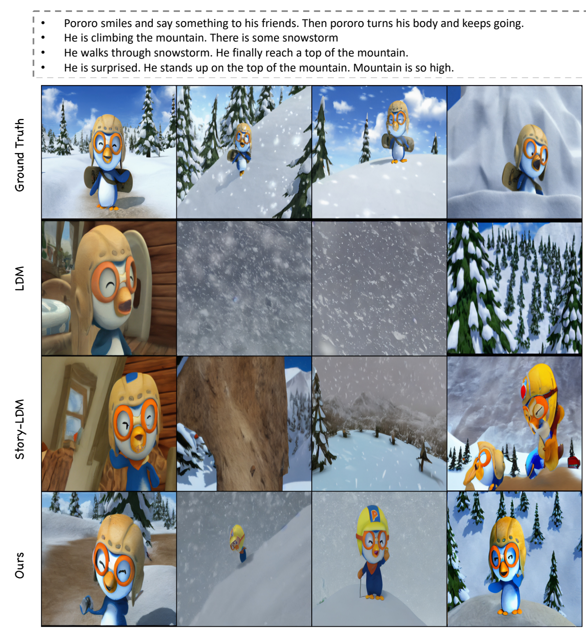

Figure 14. Qualitative comparison on PororoSV [7] with co-reference descriptions.

[7] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6329–6338, 2019. 3, 5, 7, 9

[8] Adyasha Maharana, Darryl Hannan, and Mohit Bansal. Storydall-e: Adapting pretrained text-to-image transformers for story contin-

uation. In *European Conference on Computer Vision*, pages 70–87. Springer, 2022. 3

[9] OpenAI. Dall-e 3. https://openai.com/dall-e-3/, 2023. 5, 6

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 3

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, A. 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 *ICML*, 2021. 4

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[13] Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2493–2502, 2023. 3, 4

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 4

[15] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2, 3

[16] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 3

[17] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 2, 3