

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 ANIMESHOOTER: A MULTI-SHOT ANIMATION DATASET FOR REFERENCE-GUIDED VIDEO GENERATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent advances in AI-generated content (AIGC) have significantly accelerated animation production. To produce engaging animations, it is essential to generate coherent multi-shot video clips with narrative scripts and character references. However, existing public datasets primarily focus on real-world scenarios with global descriptions, and lack reference images for consistent character guidance. To bridge this gap, we present **AnimeShooter**, a reference-guided multi-shot animation dataset. AnimeShooter features comprehensive hierarchical annotations and strong visual consistency across shots through an automated pipeline. Story-level annotations provide an overview of the narrative, including the storyline, key scenes, and main character profiles with reference images, while shot-level annotations decompose the story into consecutive shots, each annotated with scene, characters, and both narrative and descriptive visual captions. Additionally, a dedicated subset, AnimeShooter-audio, offers synchronized audio tracks for each shot, along with audio descriptions and sound sources. To demonstrate the effectiveness of AnimeShooter and establish a baseline for the reference-guided multi-shot video generation task, we introduce AnimeShooterGen, which leverages Multimodal Large Language Models (MLLMs) and video diffusion models. The reference image and previously generated shots are first processed by MLLM to produce representations aware of both reference and context, which are then used as the condition for the diffusion model to decode the subsequent shot. Experimental results show that the model trained on AnimeShooter achieves superior cross-shot visual consistency and adherence to reference visual guidance, which highlight the value of our dataset for coherent animated video generation.

1 INTRODUCTION

The animation industry plays a pivotal role in modern entertainment and education (Wells, 2013). Recent advances in AI-generated content (AIGC) have revolutionized animation production through automated creation of complex visual narratives. Professional animation workflows necessitate the generation of coherent multi-shot video sequences that maintain visual consistency and adhere to predefined character designs. This reveals a substantial gap stemming from three fundamental limitations in existing public video datasets (Bain et al., 2021; Yang et al., 2024a; Chen et al., 2024a; Wang et al., 2023b; Ju et al., 2024; Xiong et al., 2024; Wang et al., 2023a): (1) focus on real-world scenario with easily obtainable web video content, (2) reliance on global captions inadequate for multi-shot narration, and (3) absence of reference images essential for consistent character guidance across sequential shots.

In this paper, we present **AnimeShooter**, a reference-guided multi-shot animation dataset featuring comprehensive hierarchical annotations and strong visual consistency across consecutive shots. Story-level annotations define an overall storyline, main scene descriptions, and detailed character profiles with reference images. The entire story is then decomposed into ordered consecutive shots. For each shot, the shot-level annotation specifies scene, involved characters, and detailed visual captions in both narrative and descriptive forms. AnimeShooter-audio is a subset which offers additional annotations of synchronized audio for each shot, along with audio descriptions and sound sources. The dataset is constructed using an automated curation pipeline as shown in Figure 2: we first

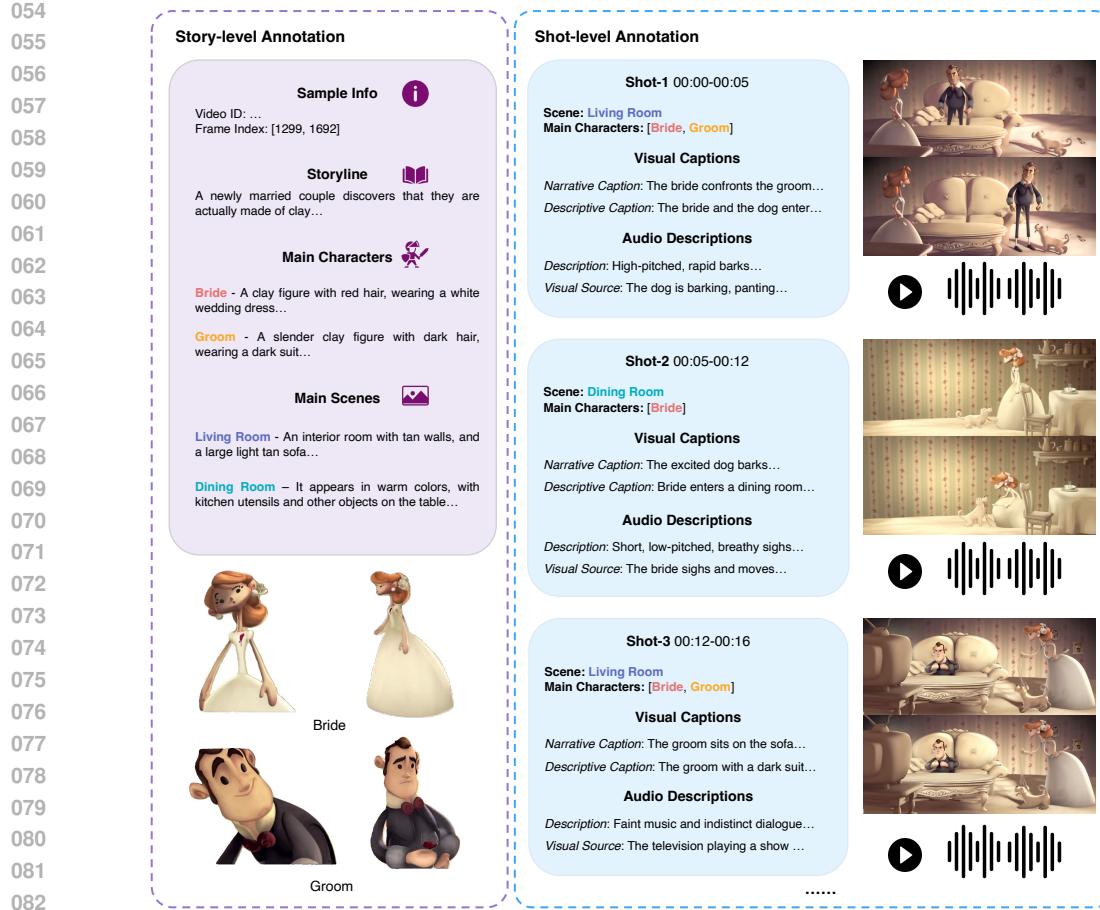


Figure 1: Overview of AnimeShooter. It is a **reference-guided multi-shot animation dataset** featuring comprehensive hierarchical annotations and strong coherence across shots. At the story level, each sample includes an overall storyline, main scene descriptions, and detailed character profiles with reference images. At the shot level, consecutive shots are annotated with specific scenes, involved characters, and rich visual captions. A specific subset, AnimeShooter-audio, additionally provides synchronized audios for each shot with corresponding audio descriptions and sound sources.

collect and filter a diverse range of large-scale animation films sourced from YouTube, then utilize Gemini (DeepMind, 2024) to generate hierarchical story scripts comprising story-level and shot-level annotations. Character reference images are extracted by sampling keyframes, segmenting characters with Sa2VA (Yuan et al., 2025) which is prompted by character ID/appearance, and ensuring quality with InternVL (Chen et al., 2024b) filtering.

To demonstrate the efficacy of AnimeShooter and establish a baseline model for this challenging task, we propose AnimeShooterGen, a reference-guided multi-shot video generation model based on MLLM and diffusion model. It can generate consecutive shots in an autoregressive manner. At each generation step, both the reference image and preceding video shots are encoded by the MLLM to produce representations that simultaneously capture character identity features and visual context. We design a multi-stage training strategy to bridge the real-to-animation domain gap and achieve autoregressive multi-shot video generation. Experiments on a custom evaluation dataset comprising multiple Intellectual Properties (IPs) and extensive evaluations demonstrate that models trained on AnimeShooter effectively learn cross-shot visual consistency and adhere to specified references.

To the best of our knowledge, this is the first reference-guided multi-shot animation dataset. Through large-scale multi-shots with visual consistency, accurate reference images for character identity, and comprehensive story and shot-level annotations, we hope AnimeShooter will facilitate research and development in narrative animation generation.

108
109

2 RELATED WORK

110
111
112
113
114
115
116
117
118
119
120

Video-Text Datasets. Existing text-to-video datasets present notable limitations for multi-shot animation generation. WebVid-10M (Bain et al., 2021) relies on readily available online video titles and primarily comprises short video clips. While dense-captioning datasets such as Vript (Yang et al., 2024a), ActivityNet (Caba Heilbron et al., 2015), Panda-70M (Chen et al., 2024a), and InternVid (Wang et al., 2023b) offer temporally segmented clips with localized descriptions, structurally akin to multi-shot annotation via timestamp-caption pairs. However, they exhibit critical narrative deficiencies, failing to maintain coherent plot progression and suffering from temporal fragmentation with abrupt gaps or redundant overlaps. In the animation domain, while efforts like AnimeCeleb (Kim et al., 2022), Sakuga-42M (Pan, 2024), and AniSora (Jiang et al., 2024a) aim to build animation datasets, they are typically restricted to single-shot content, focus narrowly on character heads, or are not publicly available, thereby limiting their utility for multi-shot animation generation.

121
122
123
124
125
126
127
128
129
130
131

Video Customization Datasets. Video customization techniques facilitate the synthesis of videos centered on specific concepts, such as individuals or objects. A critical requirement for storytelling and animation generation is multi-shot video customization: the ability to synthesize a sequence of shots that maintain the consistent appearance of a predefined character. ID-Animator (He et al., 2024) focuses on human face synthesis, utilizing facial regions as reference images. The VideoBooth dataset (Jiang et al., 2024b), derived from WebVid (Bain et al., 2021), augments textual prompts with image prompts generated by segmenting subjects from initial video frames via Grounded-SAM (Liu et al., 2024a; Kirillov et al., 2023). Many other related efforts in video customization (Chen et al., 2025; Huang et al., 2025; Deng et al., 2025; Liu et al., 2025) with similar data construction pipelines are predominantly address single-shot synthesis, and target non-animation applications. The most related multi-shot works, MovieDreamer (Zhao et al., 2024) and MovieBench (Wu et al., 2025a), are keyframe-based and limited by dataset diversity and scale respectively.

132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

Multi-Shot Storytelling and Animation Generation. Generating multi-shot videos for storytelling and animation often follows a staged pipeline. Anim-Director (Li et al., 2024) uses image generators for reference designs, which then guide keyframe generation and subsequent I2V animation. This pipeline is shared by works like VideoStudio (Long et al., 2024) and DynamisCrafter (Xing et al., 2024). Except injecting reference images via multi-modal cross-attention (e.g., Anystory (He et al., 2025), VideoStudio (Long et al., 2024)), some works attempt to maintain character appearance by optimization strategies (e.g., TaleCrafter (Gong et al., 2023), DreamRunner (Wang et al., 2024)). For example, MovieAgent (Wu et al., 2025b) leverages an LLM (Grattafiori et al., 2024; Guo et al., 2025a) for script/layout generation with ROIctrl (Gu et al., 2024) and ED-LoRA (Gu et al., 2023) for character injection. Despite these advancements, the dominant per-shot generation paradigm inherently struggles with cross-shot consistency. Recent work on Long Context Tuning (LCT) (Guo et al., 2025b) validates that autoregressive architectures can achieve enhanced holistic visual appearance and temporal coherence by recursively conditioning each shot on preceding visual contexts. But it also addresses real-world domains and faces limitations such as prohibitive training costs and a lack of explicit architectural mechanisms for reference image conditioning.

147
148

3 ANIMESHOOTER DATASET

149
150
151
152

This section describes the generation of AnimeShooter’s structured multi-shot story script and corresponding reference images. The construction pipeline is shown in Figure 2. Please refer to supplementary files for the construction of additional subset AnimeShooter-audio.

153
154

3.1 DATA COLLECTION AND FILTERING

155
156
157
158
159
160
161

Our dataset collection begins by sourcing large-scale, diverse animated content from YouTube using keywords (e.g., “short animation”, “cartoon short film”). We first filter them to ensure content relevance and minimize visual-linguistic interference. 16 uniformly sampled frames from each video are analyzed using InternVL (Chen et al., 2024b) to exclude non-animated materials (such as tutorials or film reviews) and videos containing embedded subtitles. Prolonged animated content often exhibits temporal variations in character appearance, while intricate storyline with multiple characters create cognitive overload. To mitigate these issues, we implement a duration-based filtering protocol to preserve character consistency and reduce narrative complexity. Videos exceeding 20 minutes are

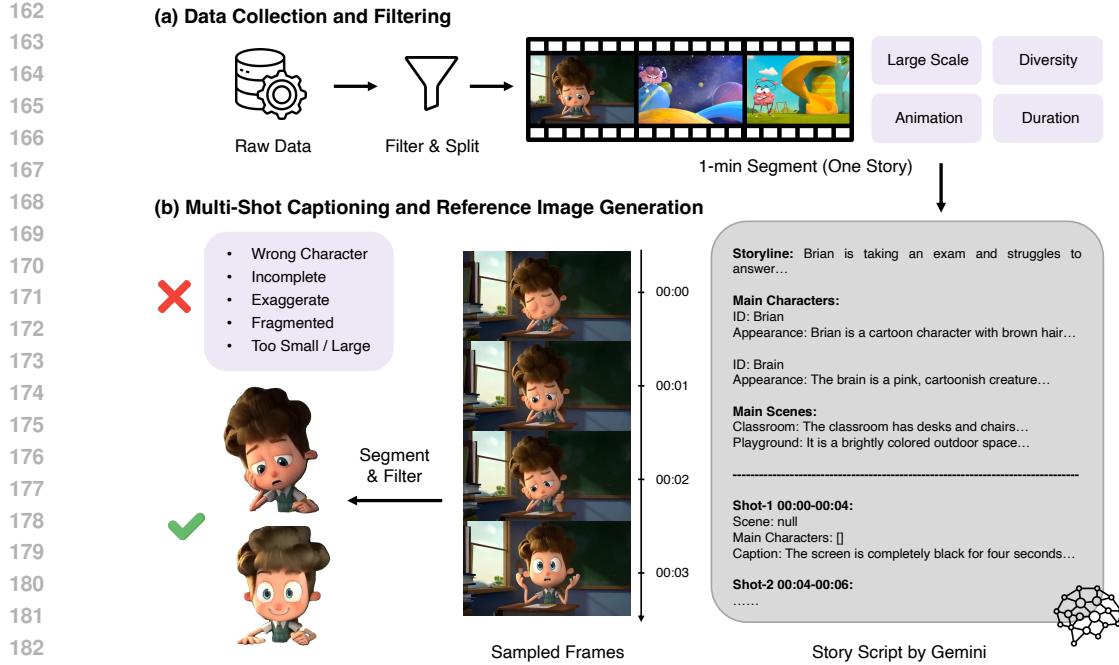


Figure 2: Video collection and annotation pipeline. We curate relevant videos from YouTube and segment them into 1-minute segments using boundary detection. Each segment serves as an individual sample representing a self-contained narrative unit (one story). We use Gemini to further decompose the story into consecutive shots with visual consistency based on transitions, and generate structured story script. Corresponding reference images are generated by Sa2VA and InternVL.

removed firstly. The remaining videos are then cut into segments with around 1-minute durations using PySceneDetect (Pys) algorithm to ensure coherent segment boundaries and narrative continuity. Each segment serves as an individual sample representing a self-contained narrative unit.

3.2 MULTI-SHOT CAPTIONING

To maintain the narrative cohesion and avoid referential ambiguity, we design a top-down multi-shot captioning strategy with three systematic phases: (1) Story-level annotation. This phase establishes a global narrative context by summarizing a succinct, coherent storyline and identifying 1-3 main characters and scenes, including detailed descriptions of their appearance and environment. (2) Shot decomposition. The entire story is subsequently decomposed into consecutive, non-overlapping shots delineated by shot transitions. (3) Shot-level annotation. For each shot, annotations identify the scene and characters, alongside two caption types: a narrative caption articulating plot progression (e.g., “The girl said goodbye to the bear”) and a descriptive caption conveying visual details (e.g., “A girl in red standing in front of a brown bear”). We utilize Gemini-2.0-flash (DeepMind, 2024) to process 1-minute segments and generate the hierarchical story script through these three phases above.

3.3 REFERENCE IMAGE GENERATION

Directly extracting frames containing a specific character is an intuitive but often inadequate strategy for obtaining reference images from animated films. The frequent co-occurrence of multiple characters and the presence of complex backgrounds can significantly hinder accurate character identification and isolation. We implement a robust model-assisted segmentation and filtering workflow. The process commences by leveraging pre-extracted story scripts to retrieve all related shots. From these shots, frames are sampled at 1 fps. Candidate frames are then fed into Sa2VA (Yuan et al., 2025), which generates initial segmentation masks based on character IDs and appearance descriptions provided as text prompts. These raw masks are refined by morphological operations to fill holes and smooth contours, contour analysis to discard masks exhibiting excessive disconnected regions, and size filtering to exclude masks that occupy less than 5% or more than 90% area of the image.

216
217 Table 1: Statistics of AnimeShooter. “Num.” for number, “Dur.” for duration, “Chars.” for characters.
218

Statistics Level	Total Num.	Avg. Dur.(s)	Avg. Caption(w)	Avg. Chars.	Avg. Scenes
Video-level	29K	286.57	-	-	-
Story-level	148K	56.72	33.55	2.26	2.20
Shot-level	2.2M	3.85	41.42	-	-

222
223 To guarantee the final quality of reference images, InternVL (Chen et al., 2024b) performs a secondary
224 verification, enforcing structural completeness of the segmented character, semantic coherence
225 between the segmented region and the provided ID/appearance prompts, consistent appearance across
226 instances relative to the character’s textual description, avoidance of frames with extreme poses or
227 expressions, and maintenance of high image resolution without motion blur.
228

229
230 3.4 DATASET STATISTICS

231 To ensure annotation fidelity within automated pipeline, we integrate human verification checkpoints
232 on a small subset, validating the accuracy of story scripts and reference images. The statistical
233 overview of the AnimeShooter dataset is presented in Table 1. The dataset contains 29K videos, each
234 with an average duration of 286.57 seconds. Videos are typically divided into 5.07 segments. Each
235 segment is approximately one minute long and serves as an individual sample representing one story.
236 These story units average 56.72 seconds and feature an average of 2.26 main characters, 2.2 main
237 scenes, and 14.82 shots. Each shot averages 3.85 seconds and is enriched with both a 10.62-word
238 narrative caption and a 30.8-word descriptive caption, summing to 41.42 words.
239

240 4 METHOD

241 To validate the utility of AnimeShooter and establish a baseline model for animation generation, we
242 introduce AnimeShooterGen. Inspired by prior works (Xiang et al., 2024; Huang et al., 2024a; Zhao
243 et al., 2024), AnimeShooterGen operates in an autoregressive fashion for reference-guided multi-shot
244 video generation. To further augment the immersive quality, we integrate AnimeShooterGen with
245 Text-to-Audio (TTA) model TangoFlux (Hung et al., 2024). As audio generation remains outside the
246 scope of this paper, please refer to Appendix D for details.
247

248 4.1 MODEL DESIGN OF ANIMESHOOTERGEN
249

250 Given the character reference image I_{ref} , the previous context of the story and a natural language
251 caption for the current shot, AnimeShooterGen predicts the current i -th video shot, denoted as S_i .
252 Figure 3 gives an overview of the model architecture. The model has two core components: the
253 autoregressive backbone stemming from a pretrained MLLM (Liu et al., 2024b) and a video generator
254 based on pretrained MMDiT (Hong et al., 2022). An adapter (Q-Former (Li et al., 2023)) is added to
255 stitch these two components. For the generation of S_i , the MLLM backbone f_{MLLM} first processes a
256 set of inputs: a reference image I_{ref} provided by user, the accumulated previous context $C_{<i}$, and the
257 textual caption T_i for the current shot. The previous context $C_{<i}$ encapsulates the long-term memory
258 from preceding shots and is composed of a visual context $V_{<i}$ and a textual context $T_{<i}$:
259

$$V_{<i} = \{F_{j,\text{end}} \mid j = 1, \dots, i-1\} \quad (1)$$

$$T_{<i} = \{T_j \mid j = 1, \dots, i-1\} \quad (2)$$

260 where $F_{j,\text{end}}$ represents the last frame of the previously generated j -th shot S_j , and T_j is its corre-
261 sponding caption. We set a sequence of learnable queries as the input of MLLM, and generate a
262 conditioning signal Cond_i :

$$\text{Cond}_i = f_{\text{MLLM}}(I_{\text{ref}}, C_{<i}, T_i) \quad (3)$$

263 This Cond_i effectively combines character visual cues from I_{ref} , long-term memory from $C_{<i}$, and
264 current textual guidance from T_i . Subsequently, the video generator synthesizes the current shot S_i ,
265 and the training objective can be formulated as follows:

$$\min_{\theta} \mathbb{E}_{t, x_0 \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0, I)} \|\epsilon - \epsilon_{\theta}(x_t, \text{Cond}_i)\|_2^2 \quad (4)$$

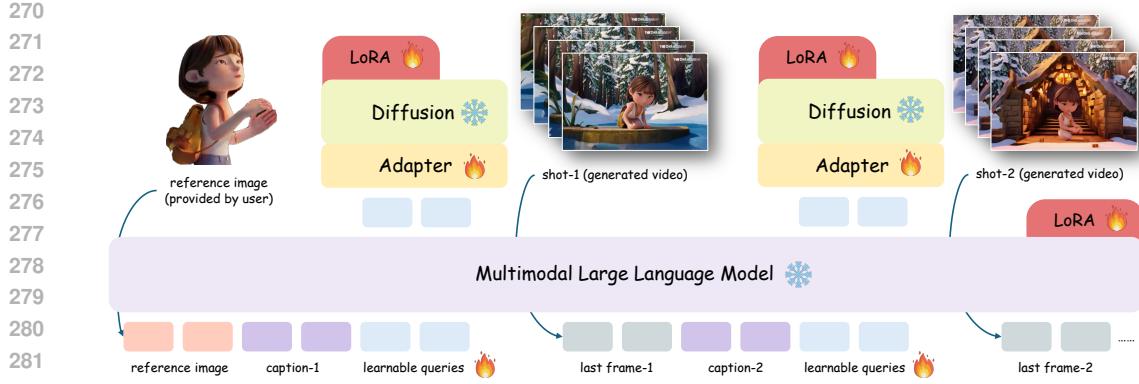


Figure 3: Overview of the model architecture. The two core components include the autoregressive backbone stemming from pretrained MLLM, and a video generator initialized from a pretrained DiT. To stitch these two components, we add a Q-Former as the adapter. This framework can generate multi-shot video in autoregressive manner.

Table 2: Quantitative comparisons of automatic metrics.

Model	Metric	Shot-level				Story-level	
		Shot-1	Shot-2	Shot-3	Shot-4	Mean	HarMeanP
IP-Adapter + I2V		0.8004	0.7814	0.7891	0.7947	0.7914	0.5901
Cogvideo-LoRA	CLIP \uparrow	0.7297	0.7200	0.7417	0.7413	0.7332	0.5028
AnimeShooterGen		0.8022	0.7949	0.7970	0.7986	0.7982	0.6121
IP-Adapter + I2V		0.3679	0.4169	0.4047	0.3870	0.3941	0.6818
Cogvideo-LoRA	DreamSim \downarrow	0.4777	0.5060	0.4842	0.4864	0.4886	0.7759
AnimeShooterGen		0.3484	0.3820	0.3799	0.3764	0.3717	0.6413

During the inference stage, upon the successful generation of shot S_i , its last frame $F_{i,\text{end}}$ and its caption T_i are incorporated into the previous context to form $C_{<i+1} = (V_{<i} \cup \{F_{i,\text{end}}\}, T_{<i} \cup \{T_i\})$, which is then used for generating the subsequent shot S_{i+1} . This autoregressive update mechanism allows the model to maintain coherence and narrative flow across multiple shots.

4.2 STAGED TRAINING

The training of AnimeShooterGen is conducted in a multi-stage fashion. The detailed training strategies and implementation details can be found in supplementary files.

Condition Alignment: The initial stage focuses on aligning the MLLM’s output conditioning signal with the text encoder of the pretrained diffusion model. MLLM processes the first frame of a ground truth video clip and corresponding caption to generate an MLLM condition. We then minimize the MSE loss between this MLLM condition and the embedding of the caption extracted from the diffusion model’s text encoder. In this stage, only the adapter and learnable queries are trainable.

Single-Shot Training: This stage aims to bridge the real-to-animation domain gap, and train MLLM to extract character visual attributes from the reference image. MLLM receives I_{ref} and T_i as input, and is optimized to produce an effective condition for generating the target shot S_i . In this stage, LoRA weights of MLLM, the adapter, and learnable queries are trainable.

Multi-Shot Training: To foster consistency across multiple shots in terms of visual appearance, style, and color palettes, this stage extends the training to sequences. MLLM now processes the reference image I_{ref} , the current caption T_i , and the previous context $C_{<i}$. LoRA weights of MLLM, the adapter, and learnable queries are trainable.

LoRA Enhancement: In all preceding stages, the core diffusion model remains frozen. To further enhance the character and style consistency provided by MLLM and refine overall video quality, this final stage involves test-time finetuning. Given a few video clips from a particular IP, we freeze all other model components and exclusively train LoRA weights added to the diffusion model.

324

325

Table 3: Quantitative comparisons of MLLM evaluation and user studies.

326

327

Model	OQ \uparrow			CRC \uparrow			MSC \uparrow			MCC \uparrow		
	GPT	Gem.	Hum.	GPT	Gem.	Hum.	GPT	Gem.	Hum.	GPT	Gem.	Hum.
IP-Adapter + I2V	6.76	6.15	2.36	7.19	5.44	2.26	6.53	6.66	3.18	6.07	5.51	2.71
Cogvideo-LoRA	6.96	4.82	2.83	6.82	2.57	2.31	6.64	4.50	2.75	6.30	3.64	2.39
AnimeShooterGen	7.19	6.88	4.23	7.87	6.54	4.72	7.15	8.24	4.63	6.68	7.07	4.52

331

332

5 EXPERIMENTS

333

5.1 EXPERIMENT SETTING

334

Baselines: We compare two mainstream methods in storytelling field. The first approach employs a short video generation model capable of IP customization to produce individual video shots. To ensure a fair comparison and highlight the benefits of our multi-shot framework, we finetune the same pretrained diffusion model, CogVideo-2B (Hong et al., 2022), on the same IP-specific dataset. The second approach first generates a series of IP-consistent keyframes, which are then transformed into video shots using an I2V model. Keyframe generation employs SDXL (Podell et al., 2023), augmented with IP-Adapter (Ye et al., 2023) to integrate reference image features, and CogVideo-5B is utilized for the I2V conversion.

344

Evaluation Dataset: We collect 20 animation films with distinct IPs. For each IP, we manually annotate 5-6 short clips for model fine-tuning. To evaluate multi-shot generation performance, we employ DeepSeek (Guo et al., 2025a) to generate 10 unique narrative prompts per IP. Each prompt describes a story comprising 4 coherent shots. This process yielded a test set of 200 stories, totaling 800 video shots.

349

Metrics: Following prior works (Wu et al., 2025b; Cheng et al., 2025; Yang et al., 2024b), we evaluate models using automatic metrics, advanced MLLM assessments and user studies. For automatic metrics, we employ CLIP score (Radford et al., 2021) and DreamSim (Fu et al., 2023) to quantify the consistency between generated characters and the reference image at shot-level and story-level. We also leverage GPT-4o and Gemini 2.5 Pro as MLLM-based judges, and conduct user studies to align with human preferences. The evaluation dimensions include Overall Quality [OQ], Character-Reference Consistency [CRC], Multi-Shot Style Consistency [MSC] and Multi-Shot Contextual Consistency [MCC]. More detailed information are in the supplementary.

357

5.2 QUANTITATIVE COMPARISONS

358

As shown in Table 2, automatic metrics evaluating character-reference alignment demonstrate that AnimeShooterGen outperforms both comparison methods. Notably, despite being trained on sequences of only 3 consecutive shots, AnimeShooterGen generalizes robustly to longer sequences during testing. Table 3 reveals additional advantages through MLLM evaluation and user studies. Beyond achieving superior CRC which also evaluates character-reference alignment, AnimeShooterGen exceeds comparison methods in MSC and MCC. Results underscore its dual strengths: (1) Enhanced reference image alignment. AnimeShooterGen achieves markedly higher character consistency than CogVideo-LoRA which shares the same diffusion architecture, proving that MLLM conditions effectively encode reference image features. (2) Cross-shot visual coherence. The MLLM’s memory mechanism retains historical context across shots, enabling high-level semantic alignment to guide the diffusion process in generating stylistically and contextually consistent new shots.

370

5.3 QUALITATIVE COMPARISONS

371

372

Figure 4 illustrates the visual outcomes of different methods on multi-shot animation generation task. The IP-Adapter + I2V approach struggles to maintain fidelity to the provided reference images due to weak control over IP-specific features. For instance, in the right-hand example, the generated character exhibits significant discrepancies in hairstyle, clothing, and facial structure compared to the reference image. CogVideo-LoRA also fails to achieve alignment with the reference images. Critically, both comparison methods generate individual shots as independent processes, leading to

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

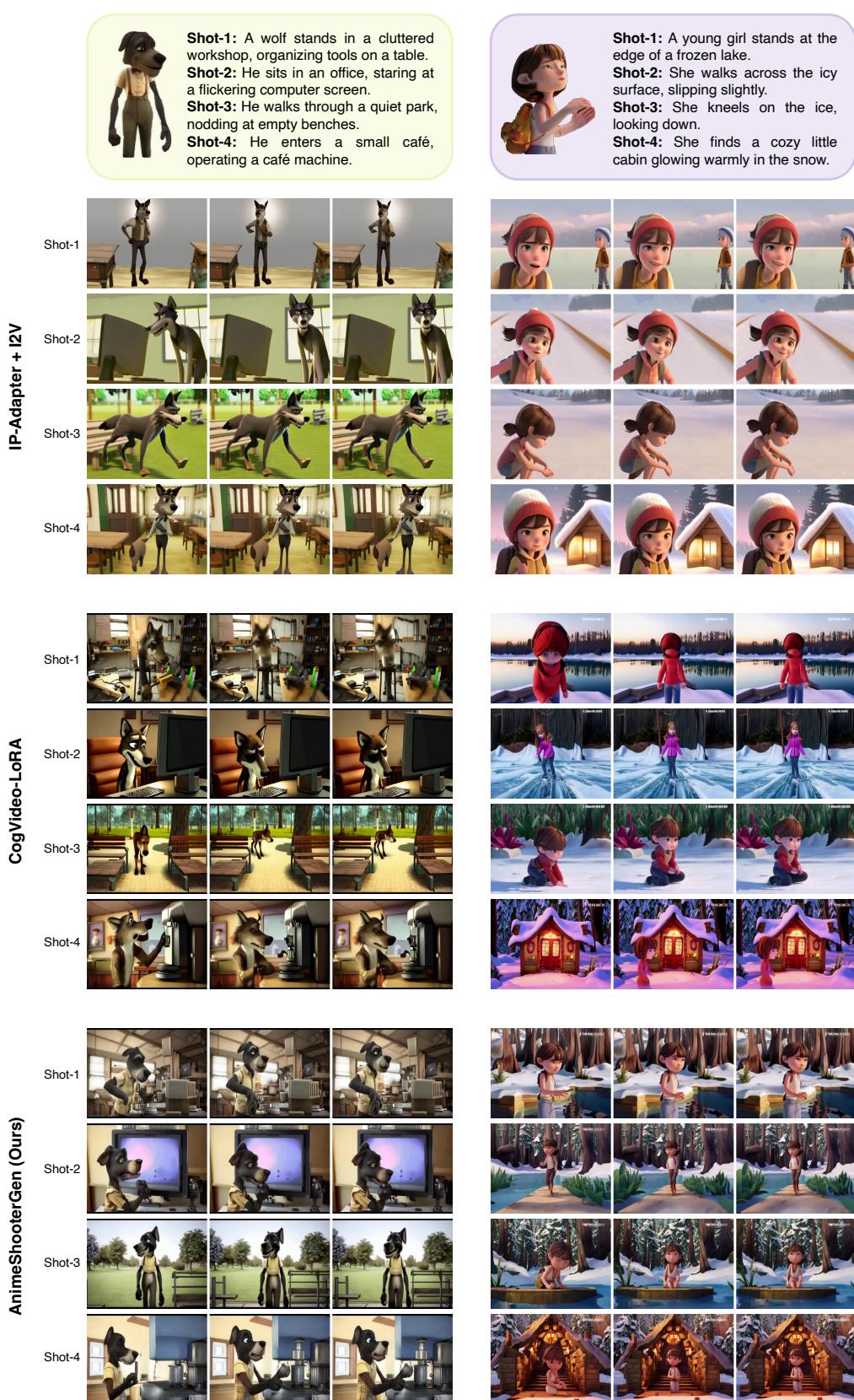


Figure 4: Qualitatively comparisons on multi-shot animation generation. Our method delivers the best visual quality, including character-reference consistency and multi-shot consistency.

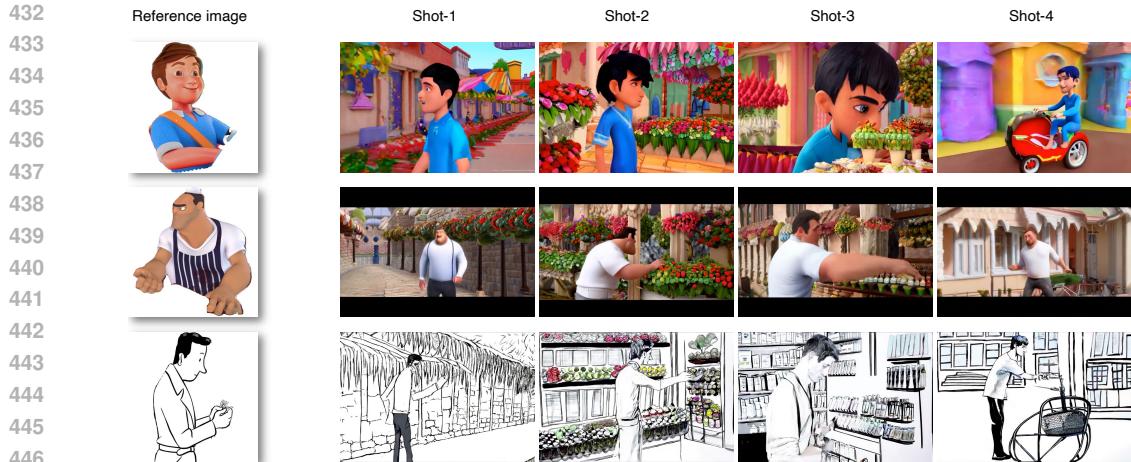


Figure 5: Visualization of using different references in MLLM (**before LoRA Enhancement**). Shared caption: “Shot-1: The man walks down a cobblestone street lined with blooming cherry trees, holding a vintage leather journal under his arm.”, “Shot-2: He pauses at a flower shop, steps inside, and begins carefully selecting flowers.”, “Shot-3: At the counter, he wraps the bouquet in paper.”, “Shot-4: He tucks the flowers into his bicycle basket and pedals away past pastel-colored storefronts.”

glaring inconsistencies between shots. In the left-hand example, both comparison methods depict the wolf in its natural animal form in the third shot, with anthropomorphic representations generated in the remaining shots. In contrast, AnimeShooterGen achieves superior reference fidelity, and sustains cross-shot consistency in style and environmental elements, as demonstrated by the invariant morphology of snow-covered trees across multiple shots. It contribute to the autoregressive generation strategy, where previously generated shots directly condition subsequent ones. This mechanism ensures robust style uniformity and contextual coherence throughout the multi-shot sequence.

5.4 INVESTIGATING THE IMPACT OF REFERENCE IMAGES

To isolate and understand the direct influence of reference images on the generation process, we omit the LoRA enhancement phase. The model is conditioned on same captions paired with distinct reference images. As illustrated in Figure 5, the reference images inject coarse-grained visual cues into the MLLM condition, influencing both character appearance and artistic style: in rows 1 and 2, the generated characters adopt clothing colors and silhouettes that closely correspond to their respective reference images; the minimalist sketch style in row 3 directly mirrors the reference image’s aesthetic. Crucially, even in the absence of LoRA enhancement, the autoregressive nature of our framework maintains strong multi-shot consistency. This observation underscores the inherent capability of the autoregressive architecture to enforce shot-to-shot coherence.

6 CONCLUSION AND LIMITATION

This paper introduces AnimeShooter, a comprehensive dataset for reference-guided multi-shot animation generation, featuring comprehensive hierarchical annotations and strong visual consistency across shots. Story-level annotations provide the storyline, key scenes, and main character profiles with reference images, while shot-level annotations decompose the story into consecutive shots, each annotated with scene, characters, and both narrative and descriptive visual captions. We also propose AnimeShooterGen which can generate reference-guided multi-shot animation in an autoregressive manner. Experiments demonstrate that being trained on AnimeShooter’s multi-shot annotations promotes cross-shot consistency and adherence to predefined references. Current limitations include the restriction of AnimeShooterGen from open-domain generation due to computational demands, the requirement for test-time fine-tuning to enhance character consistency, and suboptimal audio-visual synchronization resulting from a naive zero-shot audio generation approach. We anticipate AnimeShooter will serve as a valuable resource for future work aimed at developing more robust open-domain models with improved audio-visual alignment and character fidelity.

486 REFERENCES
487

488 Pyscenedetect. URL <https://github.com/Breakthrough/PySceneDetect>.

489 Max Bain, Arsha Nagrani, Gü̈l Varol, and Andrew Zisserman. Frozen in time: A joint video and
490 image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference*
491 *on computer vision*, pp. 1728–1738, 2021.

492 Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet:
493 A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee*
494 *conference on computer vision and pattern recognition*, pp. 961–970, 2015.

495 Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao,
496 Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m:
497 Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF*
498 *Conference on Computer Vision and Pattern Recognition*, pp. 13320–13331, 2024a.

499 Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Yuwei Fang, Kwot Sin Lee, Ivan Sko-
500 rokhodov, Kfir Aberman, Jun-Yan Zhu, Ming-Hsuan Yang, and Sergey Tulyakov. Multi-subject
501 open-set personalization in video generation. *arXiv preprint arXiv:2501.06187*, 2025.

502 Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong
503 Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning
504 for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer*
505 *Vision and Pattern Recognition*, pp. 24185–24198, 2024b.

506 Junhao Cheng, Yuying Ge, Yixiao Ge, Jing Liao, and Ying Shan. Animgamer: Infinite anime life
507 simulation with next game state prediction. *arXiv preprint arXiv:2504.01014*, 2025.

508 Google DeepMind. Gemini, 2024. URL <https://deepmind.google/technologies/gemini/>.

509 Yufan Deng, Xun Guo, Yizhi Wang, Jacob Zhiyuan Fang, Angtian Wang, Shenghai Yuan, Yid-
510 ing Yang, Bo Liu, Haibin Huang, and Chongyang Ma. Cinema: Coherent multi-subject video
511 generation via mllm-based guidance. *arXiv preprint arXiv:2503.10391*, 2025.

512 Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip
513 Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv*
514 *preprint arXiv:2306.09344*, 2023.

515 Yuan Gong, Youxin Pang, Xiaodong Cun, Menghan Xia, Yingqing He, Haoxin Chen, Longyue Wang,
516 Yong Zhang, Xintao Wang, Ying Shan, et al. Talecrafter: Interactive story visualization with
517 multiple characters. *arXiv preprint arXiv:2305.18247*, 2023.

518 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
519 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of
520 models. *arXiv preprint arXiv:2407.21783*, 2024.

521 Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao,
522 Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for
523 multi-concept customization of diffusion models. *Advances in Neural Information Processing*
524 *Systems*, 36:15890–15902, 2023.

525 Yuchao Gu, Yipin Zhou, Yunfan Ye, Yixin Nie, Licheng Yu, Pingchuan Ma, Kevin Qinghong Lin,
526 and Mike Zheng Shou. Roictrl: Boosting instance control for visual generation. *arXiv preprint*
527 *arXiv:2411.17949*, 2024.

528 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
529 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
530 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

531 Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and
532 Lu Jiang. Long context tuning for video generation. *arXiv preprint arXiv:2503.10589*, 2025b.

540 Junjie He, Yuxiang Tuo, Binghui Chen, Chongyang Zhong, Yifeng Geng, and Liefeng Bo. Anystory:
 541 Towards unified single and multiple subject personalization in text-to-image generation. *arXiv*
 542 *preprint arXiv:2501.09503*, 2025.

543

544 Xuanhua He, Quande Liu, Shengju Qian, Xin Wang, Tao Hu, Ke Cao, Keyu Yan, and Jie Zhang. Id-
 545 animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2404.15275*,
 546 2024.

547

548 Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale
 549 pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

550

551 Yuanhui Huang, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Di Zhang, Jie Zhou, and
 552 Jiwen Lu. Owl-1: Omni world model for consistent long video generation. *arXiv preprint*
 553 *arXiv:2412.09600*, 2024a.

554

555 Yuzhou Huang, Ziyang Yuan, Quande Liu, Qiulin Wang, Xintao Wang, Ruimao Zhang, Pengfei
 556 Wan, Di Zhang, and Kun Gai. Conceptmaster: Multi-concept video customization on diffusion
 557 transformer models without test-time tuning. *arXiv preprint arXiv:2501.04698*, 2025.

558

559 Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing
 560 Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video
 561 generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
 562 *Recognition*, pp. 21807–21818, 2024b.

563

564 Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan
 565 Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text
 566 to audio generation with flow matching and clap-ranked preference optimization. *arXiv preprint*
 567 *arXiv:2412.21037*, 2024.

568

569 Yudong Jiang, Baohan Xu, Siqian Yang, Mingyu Yin, Jing Liu, Chao Xu, Siqi Wang, Yidi Wu,
 570 Bingwen Zhu, Jixuan Xu, et al. Exploring the frontiers of animation video generation in the sora
 571 era: Method, dataset and benchmark. *arXiv preprint arXiv:2412.10255*, 2024a.

572

573 Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and
 574 Ziwei Liu. Videobooth: Diffusion-based video generation with image prompts. In *Proceedings of*
 575 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6689–6700, 2024b.

576

577 Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang
 578 Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured
 579 captions. *Advances in Neural Information Processing Systems*, 37:48955–48970, 2024.

580

581 Kangyeol Kim, Sunghyun Park, Jaeseong Lee, Sunghyo Chung, Junsoo Lee, and Jaegul Choo. Ani-
 582 meceleb: Large-scale animation celebheads dataset for head reenactment. In *European Conference*
 583 *on Computer Vision*, pp. 414–430. Springer, 2022.

584

585 Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete
 586 Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings*
 587 *of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

588

589 Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image
 590 pre-training with frozen image encoders and large language models. In *International conference*
 591 *on machine learning*, pp. 19730–19742. PMLR, 2023.

592

593 Yunxin Li, Haoyuan Shi, Baotian Hu, Longyue Wang, Jiashun Zhu, Jinyi Xu, Zhen Zhao, and Min
 594 Zhang. Anim-director: A large multimodal model powered agent for controllable animation video
 595 generation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

596

597 Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Qian He, and Xinglong
 598 Wu. Phantom: Subject-consistent video generation via cross-modal alignment. *arXiv preprint*
 599 *arXiv:2502.11079*, 2025.

594 Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan
 595 Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training
 596 for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer,
 597 2024a.

598 Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi,
 599 Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv*
 600 *preprint arXiv:2412.04468*, 2024b.

602 Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. Videostudio: Generating consistent-content and
 603 multi-scene videos. In *European Conference on Computer Vision*, pp. 468–485. Springer, 2024.

604 Zhenglin Pan. Sakuga-42m dataset: Scaling up cartoon research. *arXiv preprint arXiv:2405.07425*,
 605 2024.

607 Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe
 608 Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image
 609 synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

610 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
 611 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
 612 models from natural language supervision. In *International conference on machine learning*, pp.
 613 8748–8763. PMLR, 2021.

614 Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu.
 615 Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. 2023a.

617 Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan
 618 Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding
 619 and generation. *arXiv preprint arXiv:2307.06942*, 2023b.

620 Zun Wang, Jialu Li, Han Lin, Jaehong Yoon, and Mohit Bansal. Dreamrunner: Fine-grained
 621 storytelling video generation with retrieval-augmented motion adaptation. *arXiv preprint*
 622 *arXiv:2411.16657*, 2024.

624 Paul Wells. *Understanding animation*. Routledge, 2013.

625 Weijia Wu, Mingyu Liu, Zeyu Zhu, Xi Xia, Haoen Feng, Wen Wang, Kevin Qinghong Lin, Chunhua
 626 Shen, and Mike Zheng Shou. Moviebench: A hierarchical movie level dataset for long video
 627 generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp.
 628 28984–28994, 2025a.

630 Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot
 631 planning. *arXiv preprint arXiv:2503.07314*, 2025b.

632 Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao,
 633 Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions
 634 and video states. *arXiv preprint arXiv:2406.09455*, 2024.

636 Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu,
 637 Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images
 638 with video diffusion priors. In *European Conference on Computer Vision*, pp. 399–417. Springer,
 639 2024.

640 Tianwei Xiong, Yuqing Wang, Daquan Zhou, Zhijie Lin, Jiashi Feng, and Xihui Liu. Lvd-2m: A
 641 long-take video dataset with temporally dense captions. *arXiv preprint arXiv:2410.10816*, 2024.

642 Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu,
 643 and Hai Zhao. Vript: A video is worth thousands of words. *Advances in Neural Information*
 644 *Processing Systems*, 37:57240–57261, 2024a.

646 Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen.
 647 Seed-story: Multimodal long story generation with large language model. *arXiv preprint*
arXiv:2407.08683, 2024b.

648 Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt
649 adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.
650

651 Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi,
652 Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded
653 understanding of images and videos. *arXiv preprint arXiv:2501.04001*, 2025.

654 Canyu Zhao, Mingyu Liu, Wen Wang, Weihua Chen, Fan Wang, Hao Chen, Bo Zhang, and Chunhua
655 Shen. Moviedreamer: Hierarchical generation for coherent long visual sequence. *arXiv preprint*
656 *arXiv:2407.16655*, 2024.

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702 A DETAILS FOR DATA CURATION
703
704705 A.1 MULTI-SHOT CAPTIONING
706707 In Figure 6, we present the detailed prompt designed for multi-shot captioning (Section 3.2) with
708 Gemini-2.0-flash. Initially, Gemini-2.0-flash is prompted to establish a global narrative context
709 by summarizing a succinct, coherent storyline and identifying main characters and main scenes.
710 Following this analysis, it decomposes the whole story into consecutive shots and provides detailed
711 shot-level annotations.712
713
714 **USER:**715 Given a video, your task is to succinctly summarize the storyline and hierarchically decompose it into shots according to the
716 following guidelines:717 1. Ensure that the storyline is succinct and coherent.
718 2. You should identify the main characters and the main scenes (where the story takes place). Provide detailed descriptions of
719 the character's physical appearances, and the environment, style and color of the scenes.
720 3. The main character list should include only one to three important characters, those who appear repeatedly throughout the
721 storyline. Characters who appear only a few times and are not important to the storyline should be omitted.
722 4. Decompose this video into non-overlapping shots based on transitions. For each shot, provide timestamps that cover the
723 entire duration without skipping any part of the video. If a shot is longer than 4 seconds, in addition to the full shot duration,
724 select a coherent and complete 4-second segment that best represents the corresponding plot and caption. Make sure all the
725 start time and end time is connected between adjacent shots.
726 5. For each shot, provide a plot which focus on the action and story plot instead of visual descriptions. For example, "The girl
727 said goodbye to the bear" is preferred over "A girl in red standing in front of a brown bear".
728 6. For each shot, also provide a detailed descriptive caption of the scene/environment, characters, and their actions or
729 movements, even if these details have been mentioned in previous shots. For example, "By the bank of a river covered with
730 trees, a girl in red stand in front of a brown bear, smiling and waving to him".
731 7. The story may contain prologue and epilogue, which are usually characterized by a pure background, the appearance of the
732 story title, or a lot of text to indicate creation information. Please mark these meaningless prologue and epilogue that do not
733 advance the plot in your response.

734 The output format is JSON. Here is an example:

735 {
736 "storyline": ...,
737 "main characters": [
738 {
739 "ID": //name of character1,
740 "appearance": ...
741 },...
742],
743 "main scenes": [
744 {
745 "ID": //name of scene1,
746 "environment": ...
747 },...
748],
749 "shots": [
750 {
751 "start time": "MM:SS",
752 "end time": "MM:SS",
753 "extra 4-second segment": "MM:SS-MM:SS", //optional, only if the shot is longer than 4 seconds
754 "is_prologue_or_epilogue": prologue / epilogue, //optional, only if the shot is prologue or epilogue
755 "main characters": [//list IDs of main characters who appear in the current shot
756 "scene": //write ID of the scene of the current shot
757 "plot": //write the story plot
758 "caption": //write the detailed descriptive caption
759 },...
760]
761 }
762
763 <VIDEO>764 Figure 6: The prompt used for multi-shot captioning.
765

756 A.2 REFERENCE IMAGE GENERATION
757

758 To address the challenge of ambiguous segmentation in images where multiple characters or non-
759 target individuals are present, we use the prompt template for Sa2VA (Figure 7, top subfigure) that
760 explicitly incorporates character IDs and appearance descriptions from the shot-level character list,
761 followed by a clear specification of the target ID for segmentation. For post-processing, we first
762 apply morphological opening and closing operations with a kernel size of 5 to smooth boundaries and
763 remove noise, and then eliminate masks containing over 15 contours or 5 disconnected components.
764 Finally, we retain only the largest connected region, and discard masks occupying less than 5% or
765 exceeding 90% of the total image area. The prompt shown in the bottom subfigure of Figure 7 is
766 provided to InternVL for the secondary verification.

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

USER:

<IMAGE>

This image may contain characters as follows:

1. Girl, A young girl with short dark hair, wearing a pink bow.
2. Boy, A toddler with short dark hair, wearing a yellow overall.
3. Caregiver, An adult male with brown hair, a mustache, and glasses.

Please segment Caregiver.

USER:

You are tasked with evaluating whether an image can serve as a proper character reference image. Given an image and a character description, analyze the following criteria:

1. Character Completeness:

- The character is segmented from video frames, the segmented character should be complete, not fragmented
- The character's key features and details should be clearly visible

2. Description-Image Consistency:

- Compare the character's appearance in the image with the provided description
- Check if physical attributes (hair color, clothing, body type, etc.) match
- Verify if distinctive features mentioned in the description are present
- Note any inconsistencies or missing elements

3. Text-based Appearance Consistency:

- Ensure the character's appearance is consistent throughout the description
- Check if the description maintains the same physical attributes across different moments
- Reject cases where the description shows inconsistency (e.g., wearing red clothes at one time and blue clothes at another)

4. Pose Evaluation:

- The character should be in a neutral or natural standing pose
- Avoid images with exaggerated expressions or dynamic action poses
- The viewing angle should provide a clear understanding of the character

5. Image Quality:

- The image should be clear and sharp, not blurry
- Details should be easily distinguishable
- Avoid images with motion blur or poor resolution

Remember: A good character reference image should serve as a clear, accurate representation of the character's standard appearance and design.

Based on these criteria, provide:

- Verdict: yes/no on whether the image is suitable as a reference
- Explanation: a brief explanation of your verdict

Here is the character description:

<CHARACTER APPEARANCE>

Here is the image:

<IMAGE>

Figure 7: The prompt used for reference image generation. Top subfigure: Segmentation prompt for Sa2VA. Bottom subfigure: Filtering prompt for InternVL.

810 A.3 AUDIO ANNOTATION FOR ANIMESHOOTER-AUDIO
811

812 This section aims to generate visual-audio annotations by addressing the inherent asynchrony between
813 modalities (e.g., audio transitions lingering beyond visual shot cuts). To ground the model’s analysis
814 and maintain consistency with pre-existing holistic information, thus avoiding redundant reference
815 image generation, video segments alongside their story-level annotations are supplied to Gemini-2.5-
816 Pro. Gemini-2.5-Pro is required to firstly exclude video segments with prolonged background music
817 or human speech. Following this, the model executes shot decomposition and shot-level annotation.
818 It performs a joint analysis of visual and auditory cues to detect optimal clip boundaries where both
819 modalities exhibit coherent transitions. For each clip, it generates: (1) two types of visual captions,
820 (2) audio descriptors (categorizing sound types and describing tones), and (3) source attribution
821 (mapping sounds to visual elements). Notably, these clips do not strictly adhere to single visual shot
822 boundaries, as decomposition is determined by the joint consideration of both visual and auditory
823 transitions. For terminological consistency, these audio-visually coherent clips are still referred to as
824 ‘shots’. The prompt used for audio annotation is shown in Figure 8.

825 A.4 DATASET EXAMPLE
826

827 Figure 9 presents an example from AnimeShooter, featuring various scenes such as Luna’s room,
828 birthday table, and shoemaker shop, which are further divided into finer shots based on cuts.
829

830 B DETAILS FOR MODEL TRAINING
831832 B.1 MODEL FRAMEWORK
833

834 AnimeShooterGen leverages NVILA-8B (Liu et al., 2024b) as the pretrained MLLM backbone and
835 CogVideo-2B (Hong et al., 2022) as the pretrained video diffusion model. An adapter, crucial for inter-
836 model communication, is implemented using a QFormer (Li et al., 2023) with 12 layers. The length of
837 learnable queries is set to 64. In AnimeShooterGen, for a sequence including n shots, MLLM receives
838 an input formatted as “<Reference><Caption₁><LQ><Frame₁>...<Caption_n><LQ>”,
839 where each <Caption_i> represents textual guidance for the i -th shot and <LQ> serves as a place-
840 holder of learnable query for contextual feature extraction. During training, the <Frame₁> to
841 <Frame_{n-1}> tokens are populated with the last frames from their corresponding ground truth video
842 shots, and all n shots contribute jointly to the diffusion loss through backpropagation. At inference
843 time, the model operates autoregressively: it first generates a 1-shot sequence using only the initial
844 reference, then replaces the <Frame₁> token with the final frame of the newly generated shot to
845 condition the next iteration, recursively extending the sequence until reaching the target length n .
846

847 B.2 IMPLEMENTATION DETAILS
848

849 For Condition Alignment, we train the model using video-caption pairs from the large-scale WebVid-
850 10M dataset (Bain et al., 2021). MLLM takes the first frame of a ground-truth video clip, its
851 corresponding caption, and learnable queries as inputs. We minimize the MSE loss between the
852 MLLM’s output condition and the T5 features of the caption. During this phase, only the adapter and
853 learnable queries are trainable, optimized with a batch size of 32 for 1.8 M steps.
854

855 For Single-Shot Training, we utilize samples from the AnimeShooter dataset containing reference
856 images and descriptive captions for individual shots. Here, the adapter, learnable queries, and LoRA
857 parameters of the MLLM are fine-tuned with a batch size of 24 for 17K steps. Classifier-Free
858 Guidance (CFG) is applied to enhance multimodal control, with independent dropout probabilities of
859 0.05 for the reference image, caption, or both modalities.

860 For Multi-Shot Training, we curate 3-shot sequences from the AnimeShooter dataset. The same
861 trainable components (adapter, learnable queries, and MLLM LoRA) are updated with a batch
862 size of 8 for 8K steps. A simplified CFG strategy is adopted, where both the reference image and
863 caption inputs are simultaneously replaced with blank content at a 0.05 probability, eliminating
modality-specific dropout.

```

864
865 USER:
866 Analyze the input animation video and perform visual-audio annotation following these steps:
867 # Video Verification
868 Ensure the video does not contain extended segments of background music (BGM) or human speech. The audio of this video
869 should feature only sound effects (e.g., footsteps, alarms, animal sounds, environmental sounds).
870 # Shot Segmentation
871 Split the video into non-overlapping multi-shot structure using combined visual+audio boundaries where:
872 - Make sure all the start time and end time is connected between adjacent shots.
873 - Each shot maintains continuous visual narrative and audio context.
874 - Transition points must show simultaneous visual+audio changes.
875 - Each shot lasts more than 3 seconds.
876 - For each shot, provide start/end timestamps (MM:SS).
877 # Shot Annotation
878 Based on the given video-level annotation (main characters, main scenes and storyline), you should create shot-level
879 annotations for each shot:
880 (1) List main characters present in this shot and the scene of this shot.
881 (2) The video may contain prologue and epilogue, which are usually characterized by a pure background, the appearance of the
882 story title, or a lot of text to indicate creation information. Please mark these meaningless prologue and epilogue that do not
883 advance the plot in your response.
884 (3) Visual annotation: For each shot, provide a plot which focus on the action and story plot instead of visual descriptions. For
885 example, "The girl said goodbye to the bear" is preferred over "A girl in red standing in front of a brown bear". For each shot,
886 also provide a detailed descriptive caption of the scene/environment, characters, and their actions or movements, even if these
887 details have been mentioned in previous shots. For example, "By the bank of a river covered with trees, a girl in red stand in
888 front of a brown bear, smiling and waving to him".
889 (4) Audio annotation: For each shot, provide three description layers for only sound effects. The first is "type", it means sound
890 effect category, e.g., alarm, magical plink, scampering. The second is "sound description", it means technical audio
891 characteristics (pitch, intensity, temporal pattern, etc.). The third is "visual context", it means on-screen source/trigger of sound.
892
893 # Output Format
894 Structure as hierarchical JSON. If the video contains extended segments of background music (BGM) or human speech, return
895 empty JSON object.
896 json
897 {
898   "storyline": ...,
899   "main characters": [
900     {
901       "ID": ...,
902       "appearance": ...
903     },
904   ],
905   "main scenes": [
906     {
907       "ID": ...,
908       "environment": ...
909     },
910   ],
911   "shots": [
912     {
913       "start time": "MM:SS",
914       "end time": "MM:SS",
915       "is_prologue_or_epilogue": prologue / epilogue, //optional, only if the shot is prologue or epilogue
916       "main characters": [ //list IDs of main characters who appear in the current shot],
917       "scene": ..., //write ID of the scene of the current shot
918       "visual annotation": {
919         "plot": ...,
920         "caption": ...
921       },
922       "audio annotation": {
923         "type": ...,
924         "sound description": ...,
925         "visual context": ...
926       }
927     },
928   ],
929 }
930
931 # Now, please analyze the following case.
932 <VIDEO>
933 <VIDEO LEVEL ANNOTATION>

```

Figure 8: The prompt used for constructing AnimeShooter-audio.

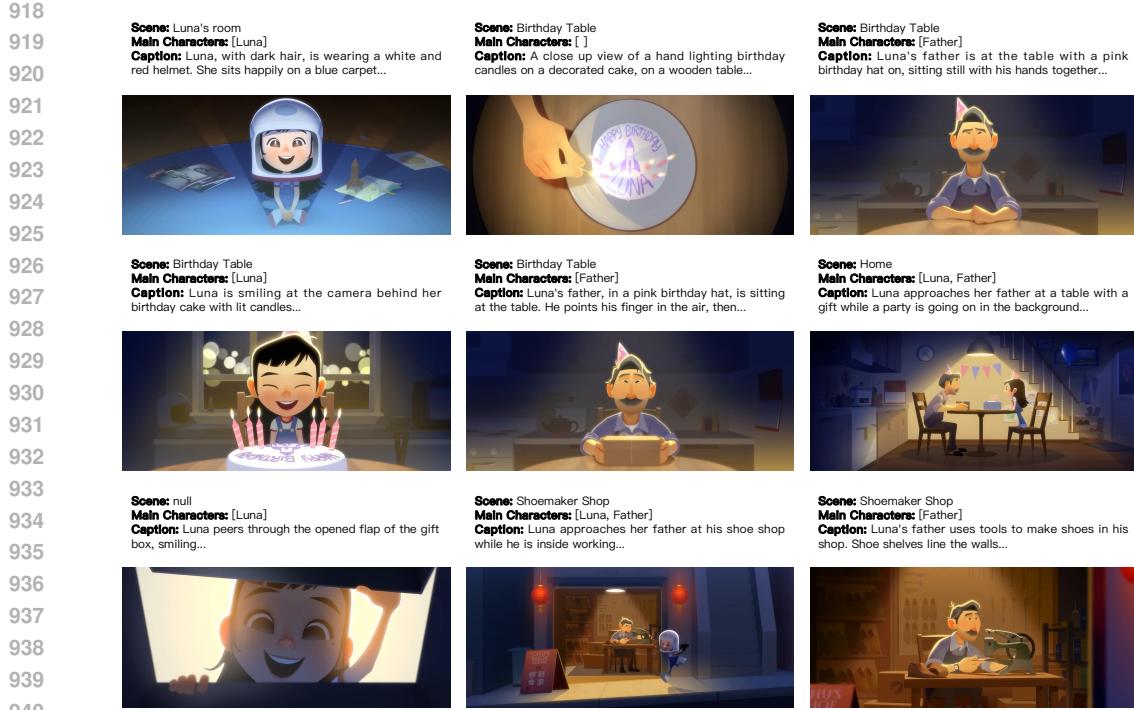


Figure 9: Multi-shot dataset example with different scenes.

943 For LoRA enhancement, given 5-6 separate video clips from a particular IP, these are randomly
 944 sampled and combined to form training sequences of 3 consecutive shots. This targeted finetuning is
 945 performed for 1K to 2K steps with batch size of 2.

946 All training stages are implemented on GPUs with 40G memory. Here are computational costs of
 947 different training stages: Condition Alignment for 1400 GPU hours; Single-Shot Training for 900
 948 GPU hours; Multi-Shot Training for 400 GPU hours; LoRA Enhancement for 20 GPU hours. The
 949 inference can be implemented on a single GPU with 40G memory with pipeline offloading. The
 950 inference time of a story with 4 shots is around 720s.

C DETAILS FOR EXPERIMENTS

C.1 BASELINES

957 For fair comparison with CogVideo-LoRA, we fine-tune the same pretrained model (CogVideo-2B)
 958 on the same IP-specific dataset and iterations as AnimeShooterGen. While AnimeShooterGen is
 959 trained in multi-shot mode (batch size=2 per step, with each sample containing 3 shots, totaling 6
 960 shots per step), CogVideo-LoRA supports only single-shot training. To match the computational
 961 scale, we set its batch size to 6 (6 shots per step). For the training-free baseline IP-Adapter + I2V,
 962 we utilize stable-diffusion-xl-base-1.0 with its IP-Adapter to generate keyframes conditioned on
 963 reference images and captions. These keyframes and captions are then fed to the CogVideo-5B I2V
 964 model (replacing CogVideo-2B due to its lack of I2V capability) to synthesize video shots.

C.2 EVALUATION DATASET CONSTRUCTION

967 We construct a manually annotated evaluation dataset of 20 animated films featuring distinct IPs to
 968 support LoRA enhancement for AnimeShooterGen and finetuning for CogVideo-LoRA. For each IP,
 969 alongside a reference image, we curate 5-6 short video clips (each lasting several seconds) exclusively
 970 depicting the main character, ensuring maximal diversity in actions and scenes, as shown in Figure
 971 10. To evaluate multi-shot generation performance, we employ DeepSeek with prompt shown in
 Figure 11 to generate 10 unique narrative prompts per IP.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025



Figure 10: Example of IP-specific dataset for evaluation.

USER:

Generate 10 prompts for evaluation cases. Each prompt must contain 4 consecutive scenes with simple captions following these rules:

1. Storytelling format with scene continuity but distinct background changes and different events.

- Fixed character reference (it will be provided)
- Only one character in all scenes

2. Keep captions simple.

- Use basic verbs and no complex interactions.
- Use simple language without complicated adjectives or descriptive phrases.
- Without small objects or detailed props.

3. Follow this exact JSON format:

```
{
  "prompts": [
    [
      "scene1 description",
      "scene2 description",
      "scene3 description",
      "scene4 description"
    ],
    // 9 more prompts
  ]
}
```

Now, please generate prompts for a young girl.

Figure 11: The prompt used for constructing evaluation dataset.

1026
1027

C.3 AUTOMATIC METRICS

1028
1029
1030
1031
1032
1033
1034
1035

We employ CLIP score and DreamSim to quantify the visual similarity between generated characters and the reference image. We uniformly sample 5 frames from each shot and compute the average similarity score as the shot-level metric. To assess story-level consistency across 4 shots, we introduce two metrics: (1) Mean Similarity (Mean), the arithmetic mean of the 4 shot-level similarity scores. (2) Penalized Harmonic Mean Similarity (HarMeanP). Recognizing that even a single poorly generated shot can disrupt viewer immersion, this metric penalizes the worst-performing shot. This metric first calculates the harmonic mean of all 4 shots' similarity scores (chosen for its sensitivity to extremely low values), then multiplies this result by the lowest similarity score as an additional penalty term.

1036
1037

C.4 MLLM ASSESSMENT

1038
1039
1040
1041
1042

We leverage GPT-4o and Gemini 2.5 Pro as MLLM-based judges. To mitigate ordering bias in evaluation, we employ the prompt template shown in Figure 12 and change the presentation order across three independent evaluation rounds. The final results reported in Section 5.2 represent the averaged metrics from these three trials, ensuring robustness against positional preferences. MLLM uses 1-10 scoring (1=worst, 10=best) with one-point increments. Evaluation dimensions including:

1043
1044
1045
1046
1047
1048
1049
1050
1051
1052

- Overall quality [OQ]. A holistic assessment considering aesthetic appeal, image quality, visual consistency and so on.
- Character-Reference Consistency [CRC]. Visual fidelity of the generated character to the provided reference image.
- Multi-Shot Style Consistency [MSC]. Coherence of artistic style, color palette, and texture across all shots within a story.
- Multi-Shot Contextual Consistency [MCC]. Continuity of the narrative and context across shots, e.g., ensuring the character maintains a consistent appearance appropriate to the unfolding story.

1053

C.5 HUMAN EVALUATION

1054
1055
1056
1057
1058
1059
1060
1061

For the human evaluation, we recruit 10 participants who hold at least a bachelor's degree and have prior experience in image or video generation. A total of 15 stories with 60 shots are presented to the participants. The evaluation metrics align with those defined in Section C.4. Participants are instructed to score each dimension (1: lowest, 5: highest) based on the reference image, corresponding captions, and multi-shot videos generated by the three models. The final performance of each model is calculated as the average scores across all responses.

1062
1063

C.6 MODEL PERFORMANCE ON LONGER STORYLINES

1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

To investigate the performance of AnimeShooterGen on longer sequences, we utilize DeepSeek to generate longer stories comprising 15 shots. To manage computational demands, we employ a sliding window inference strategy for shots beyond position 4: each shot i is generated using the previous three shots ($i - 3, i - 2, i - 1$) as its context. These 15-shot sequences are divided into three segments (shots 1-5, 6-10, and 11-15). We present these segments to GPT and Gemini in three different orders, minimizing potential position bias. We also calculate CLIP and DreamSim metrics. Results shown in Table 4 reveal that the model maintains relatively robust performance on extended sequences, but still has slight performance degradation due to error accumulation. **We show qualitative results in Figure 13. While quantitative results indicate a slight decline in character similarity over extended sequences, this degradation is hardly noticeable in qualitative assessments, further demonstrating the robustness of the method in temporal extension. However, we also observe other challenges arising from long sequences, such as increased distortion in backgrounds (first row) and characters (second row), as well as a gradual drift in color tone (last row).**

C.7 ADDITIONAL QUALITATIVE RESULTS

We present additional qualitative results in Figure 14 and Figure 15. Both comparison methods exhibit limitations in preserving character consistency with the reference image and cross-shot coherence. As

```

1080
1081 USER:
1082 Please act as an impartial judge and evaluate three AI-generated multi-shot video samples (4 shots each) based on the
1083 following inputs and requirements.
1084
1085 # Input
1086 1. Character reference image (provided first)
1087 2. Text captions for each of the 4 shots (provided second)
1088 3. Three models' outputs - Each model's output is a 4-row grid, where:
1089 Each row corresponds to one shot (Shot 1 to Shot 4 from top to bottom).
1090 Within each row, 3 sampled frames are placed horizontally (left to right).
1091
1092 # Evaluation Dimensions
1093 Use 1-10 scoring (1=worst, 10=best) with one-point increments allowed:
1094
1095 1. Overall Quality <OQ>
1096 Conduct a comprehensive assessment of the overall quality for each model-generated sample, evaluating multiple dimensions
1097 such as visual consistency, aesthetic appeal, image quality, and other relevant factors.
1098
1099 2. Character-Reference Consistency <CRC>
1100 Evaluate visual fidelity and consistency between generated characters and the provided reference image.
1101
1102 3. Multi-Shot Style Consistency <MSC>
1103 Assess style, color palette and texture coherence across all shots.
1104
1105 4. Multi-Shot Contextual Consistency <MCC>
1106 Verify contextual continuity. E.g., evaluate whether the character maintain a stable appearance, behavior, and role throughout
1107 shots.
1108
1109 # Requirements & Output Format
1110 Avoid any bias, ensure that the order of presentation does not affect your decision. Do not favor certain agent names. Reflect
1111 the score differences as much as possible.
1112
1113 Return JSON format with detailed feedback:
1114 {
1115   "Model_A": {
1116     "scores": {
1117       "OQ": 4,
1118       "CRC": 7,
1119       "MSC": 3,
1120       "MCC": 1
1121     },
1122     "feedback": {
1123       "strengths": ["Consistent eye detail", "Good lighting continuity"],
1124       "weaknesses": ["Hand proportions vary in Shot 3", "Background mismatch in Shot 4"]
1125     }
1126   },
1127   "Model_B": {...},
1128   "Model_C": {...},
1129   "comparative_analysis": "Model A excels in character consistency and performs best..."
1130 }
1131
1132
1133

```

Figure 12: The prompt used for MLLM assessment.

Table 4: Model Performance on Longer Storylines.

Shots	CLIP \uparrow	DreamSim \downarrow	OQ \uparrow		CRC \uparrow		MSC \uparrow		MCC \uparrow	
			GPT	Gem.	GPT	Gem.	GPT	Gem.	GPT	Gem.
Shot 1-5	0.7989	0.3637	7.19	6.99	7.94	7.59	6.84	7.34	6.45	6.37
Shot 6-10	0.7945	0.3778	6.83	6.46	7.67	7.26	6.52	7.21	6.01	5.51
Shot 11-15	0.7961	0.3746	6.96	6.52	7.76	7.08	6.64	7.11	6.18	5.61

illustrated in the right panel case of Figure 15, the IP-Adapter+I2V framework erroneously transforms the flashlight-equipped helmet into a color-mismatched hat. In the left panel case, the first shot generated by CogVideo-LoRA demonstrates a visually discordant art style. In contrast, our proposed AnimeShooterGen achieves superior preservation of character identity, color palette continuity, and stylistic consistency across generated sequences.

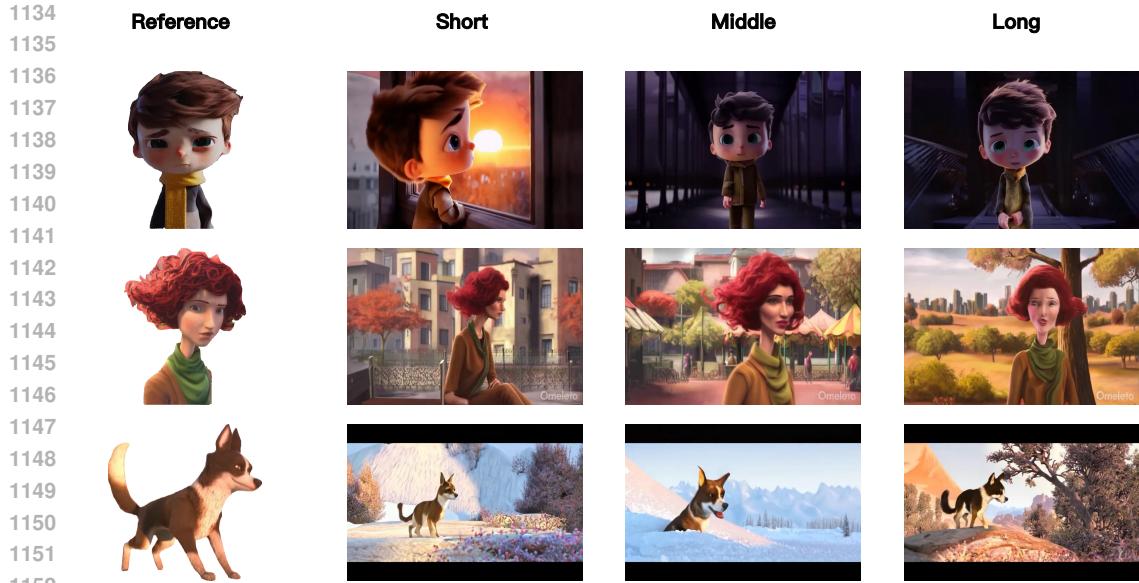


Figure 13: Qualitative results for longer storylines.

Table 5: Results of evaluation on VBench.

Model	Subject Cons.	Bg Cons.	Motion Smth.	Dyn. Degree	Aesthetic	Imaging
IP-Adapter + I2V	0.9341	0.9543	0.9873	0.6500	0.5959	0.6073
Cogvideo-LoRA	0.9330	0.9539	0.9904	0.7025	0.6171	0.6524
AnimeShooterGen	0.9366	0.9517	0.9921	0.7124	0.6255	0.6512

C.8 VIDEO QUALITY

Table 5 presents the evaluation results on VBench (Huang et al., 2024b). Although IP-Adapter + I2V employs the more powerful CogVideo-5B model, it does not demonstrate superior video quality, likely due to the domain gap between real-world and animation. In contrast, CogVideo-LoRA and AnimeShooterGen which are trained on animation dataset achieve comparable or even better quality with only a 2B model, and AnimeShooterGen achieves the best performance across most metrics.

C.9 ABLATION STUDIES

Table 6 presents the results of ablation study, which includes the following configurations: (1) w/o MLLM condition, where the MLLM condition is replaced with the original text condition (i.e., CogVideo-LoRA); (2) w/o LoRA Enhancement, where no LoRA fine-tuning based on a specific IP, and zero-shot capability is tested directly; and (3) w/o Reference Image, where the reference image in the MLLM is replaced with a blank image. The results further confirm that after training on the AnimeShooter dataset, the MLLM is able to capture high-level semantic consistency. Replacing it with the diffusion model’s original text condition or using a blank image significantly degrades model performance. Moreover, the results highlight the importance of the detail alignment achieved during the LoRA Enhancement stage, and the high-level semantic consistency provided by the MLLM cannot be fully realized at the pixel level without it.

C.10 IMPACT OF DIFFERENT VISUAL CONTEXT FRAMES

This section examines how the choice of different frames as the visual context influences model performance. We present in Table 7 the results under different configurations: using the first frame (first), the middle frame (mid), the last frame (last), these three frames simultaneously (multi), and the pooled feature of them (pool). The results demonstrate that the choice of frame has no notable impact on model performance. Moreover, incorporating additional frames or using feature pooling



Figure 14: Additional qualitative results.

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

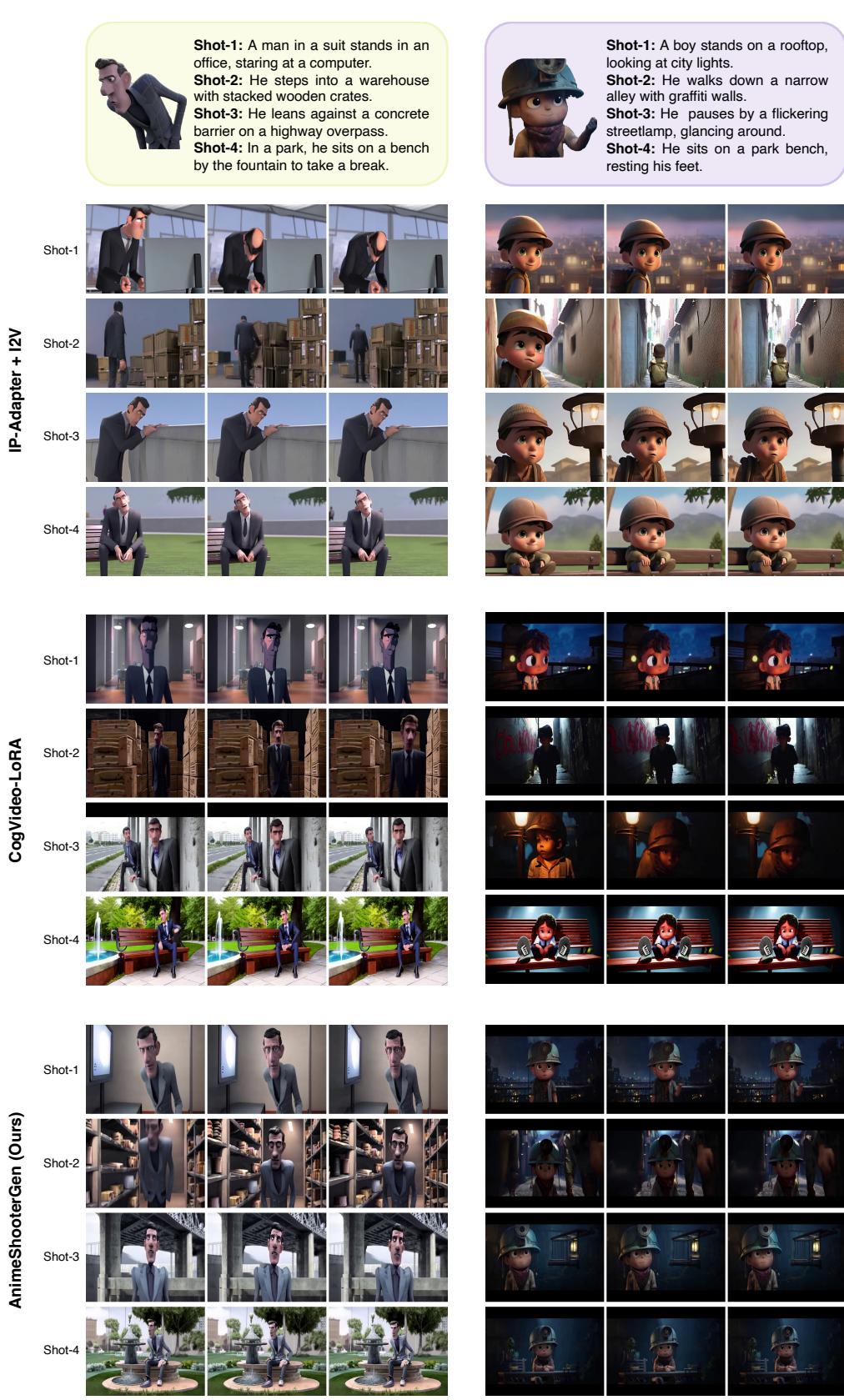


Figure 15: Additional qualitative results.

1296

1297

Table 6: Results of ablation studies.

Model	Metric	Shot-level				Story-level	
		Shot-1	Shot-2	Shot-3	Shot-4	Mean	HarMeanP
w/o MLLM condition	CLIP \uparrow	0.7297	0.7200	0.7417	0.7413	0.7332	0.5028
w/o LoRA Enhancement		0.7628	0.7430	0.7376	0.7362	0.7449	0.5280
w/o Reference Image		0.7065	0.7167	0.7208	0.7264	0.7176	0.4934
AnimeShooterGen		0.8022	0.7949	0.7970	0.7986	0.7982	0.6121
w/o MLLM condition	DreamSim \downarrow	0.4777	0.5060	0.4842	0.4864	0.4886	0.7759
w/o LoRA Enhancement		0.4773	0.5198	0.5178	0.5323	0.5118	0.7940
w/o Reference Image		0.5363	0.5348	0.5213	0.5131	0.5264	0.7926
AnimeShooterGen		0.3484	0.3820	0.3799	0.3764	0.3717	0.6413

1308

1309

Table 7: Impact of different visual context frames.

Visual context frame	Metric	Shot-level				Story-level	
		Shot-1	Shot-2	Shot-3	Shot-4	Mean	HarMeanP
first	CLIP \uparrow	0.8152	0.8017	0.8068	0.8115	0.8088	0.6286
mid		0.8108	0.8033	0.7992	0.8113	0.8062	0.6238
last		0.8168	0.8055	0.8116	0.8094	0.8108	0.6323
multi		0.8103	0.7995	0.8059	0.8031	0.8047	0.6223
pool		0.8166	0.8007	0.8007	0.8025	0.8051	0.6205
first	DreamSim \downarrow	0.3318	0.3697	0.3639	0.3591	0.3561	0.6205
mid		0.3304	0.3676	0.3692	0.3599	0.3568	0.6222
last		0.3325	0.3714	0.3625	0.3590	0.3563	0.6226
multi		0.3344	0.3758	0.3740	0.3750	0.3648	0.6334
pool		0.3355	0.3809	0.3701	0.3652	0.3629	0.6307

1323

1324 **does not lead to performance improvement. Considering computational efficiency, a single frame**
 1325 **suffices as the visual context, as it already contains sufficient visual information.**

1326

1327

D INTEGRATION OF AUDIO GENERATION CAPABILITIES FOR ANIMESHOOTERGEN

1330

1331 To further augment the immersive quality, we integrate AnimeShooterGen with zero-shot Text-to-
 1332 Audio (TTA) generation using TangoFlux (Hung et al., 2024). The workflow involves processing the
 1333 video captions and keyframes with GPT-4o to generate descriptive audio captions with the prompt
 1334 shown in Figure 16. These audio captions subsequently guide TangoFlux in synthesizing audio tracks,
 1335 which are then merged with the video sequences. However, results reveal substantial limitations
 1336 in current simplistic zero-shot audio generation paradigms. Primarily, the decoupled generation
 1337 processes for visual and auditory modalities result in inherent inter-modality synchronization failures.
 1338 For example,脚步 sounds lag behind walking animations, or character facial expressions mismatch
 1339 with voice. Furthermore, constrained by existing text-to-audio models’ performance, environmental
 1340 sound effects such as gentle breezes, engine roars, and mechanical hums fail to achieve sufficient
 1341 perceptual distinctiveness, thereby compromising the immersive experience. We propose an audio-
 1342 annotated subset named AnimeShooter-audio, hoping to facilitate and encourage future research
 1343 into the development of more sophisticated audio-visual co-generation models capable of achieving
 1344 tighter synchronization and semantic coherence.

1344

1345

E POTENTIAL SOCIAL IMPACTS

1347

1348

1349 While generation models democratize content creation, they risk enabling malicious applications such
 as generating deepfakes for disinformation and producing harmful content. To address these risks, the
 research community and policymakers must adopt proactive safeguards. Technical countermeasures

```

1350
1351 USER:
1352 Analyze the three provided animation frames and their corresponding text caption. Generate a descriptive audio prompt for a
1353 text-to-audio model to create corresponding ambient sounds.
1354
1355 # Requirements:
1356 - Prioritize sounds logically implied by the visuals and caption, e.g., dog barks if a dog is shown.
1357 - Specify the source of each sound explicitly, e.g., whirring of a coffee machine, rustling of leaves.
1358 - Sound effects may include: ambience (e.g., wind, city noise), action sounds (e.g., footsteps, door creaks)...
1359 - Avoid abstract metaphors, focus on concrete sounds a model can generate.
1360
1361 # Example Output:
1362 Only one phrase representing one type of sound. Directly output the audio prompt, no other text.
1363 For example:
1364 "a cow is mooing"
1365 "A stream flows"
1366
1367 # Generate the audio prompt now.
1368 <CAPTION>
1369 <FRAMES>
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

```

Figure 16: The prompt used for generating descriptive audio captions.

1366 include embedding watermarks in generated videos for provenance tracking, deploying AI-driven
 1367 detectors to flag synthetic content, and implementing strict filters to block unethical prompts.
 1368

1369 In light of ethical considerations, we take deliberate measures to ensure the responsible and legal
 1370 use of data. Our dataset does not involve the distribution of any copyrighted video content. We only
 1371 release binary masks and text annotations. All copyrights for videos remain with the original creators
 1372 and YouTube. We provide video IDs to allow access to the source material, and fully respect the
 1373 creators' ownership and control over their work. By focusing on the animation domain, our research
 1374 also circumvents privacy concerns associated with real-world videos.
 1375

1376 F LICENSE

1377 The AnimeShooter dataset is released under the CC BY-NC 4.0 License and the data is collected from
 1378 publicly accessible sources. The released dataset includes annotated scripts, reference image masks,
 1379 and corresponding video IDs, while the original source videos must be obtained independently from
 1380 YouTube using the provided IDs. AnimeShooterGen is built upon two pretrained models: NVILA
 1381 and CogVideo. We release the code and model weights of AnimeShooterGen under the Apache 2.0
 1382 License, and provide a copy of original licenses of NVILA and CogVideo in our GitHub repository.
 1383

1384 G THE USAGE OF LARGE LANGUAGE MODELS

1385 We utilize Large Language Models solely for language editing and proofreading of the manuscript.
 1386 All research, including ideation, literature review, and analysis, as well as the initial drafting, is
 1387 conducted by the authors without the assistance of AI.
 1388

```

1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

```