ShotAdapter: Text-to-Multi-Shot Video Generation with Diffusion Models

Supplementary Material

S.1. Full Videos

For the complete videos, please see the HTML file in the supplementary zip.

S.2. Quality Analysis

In our user study (Fig. 7), each participant views two videos simultaneously, selected from a pool of 10 randomly chosen videos from the generated results, with one video always generated by ShotAdapter. Participants are then asked to choose their preferred video based on Identity Consistency (IC), Background Consistency (BC), Text Alignment (TA), and Quality (Q). Our approach achieves superior results in a 1-to-1 comparison with baselines in Identity (IC) and Background Consistency (BC). In terms of Text Alignment (TA), it achieves a slight improvement over Similar Shots (SS) and Random Shots (RS), while outperforming Shots by Reference (SR) by a substantial margin. While our model shows slightly lower performance in Quality (Q) compared to RS and SS, this can be attributed to fine-tuning with a 90% reduced batch size, emphasizing the lightweight nature of our approach as it still brings competitive results despite a training with much reduced batch size. Moreover, our model is shown to be better than SR as using an off-the-shelf method [62] and combining it with I2V model even further propogates the error, resulting in worse quality. Despite this minor trade-off, our method excels in all other metrics, demonstrating its robustness and effectiveness for multi-shot video generation.

Although our model demonstrates slightly lower quality compared to the baselines in the user study, we hypothesize that this is due to the significantly reduced batch size during fine-tuning. Specifically, we utilize a batch size that is 90% smaller than the one used during pre-training. Table 3 illustrates how varying the batch size impacts the quality of the generated videos, measured by Frechet Video Distance (FVD) [47]. When fine-tuning with a batch size of 32, whether on the original video dataset (Default Dataset) without our data collection pipeline or on our processed dataset (ShotAdapter), the FVD scores remain comparable. Notably, our method (ShotAdapter) achieves slightly better scores, suggesting that the reduced batch size during fine-tuning, rather than our data processing approach, accounts for the measured differences in video quality metric. Furthermore, our current checkpoint, fine-tuned with a batch size of 128, shows improved quality compared to the 32 batch size setups. However, it still performs slightly worse than the original checkpoint, which was pre-trained with a significantly larger batch size (>1000).

Table 3. ***Quality comparison with varying batch size***

	32 batch size		**>1000 batch size**	**128 batch size**
	Default Dataset	*ShotAdapter*	*Default Dataset*	*ShotAdapter*
FVD	477.18	473.30	357.73	401.52

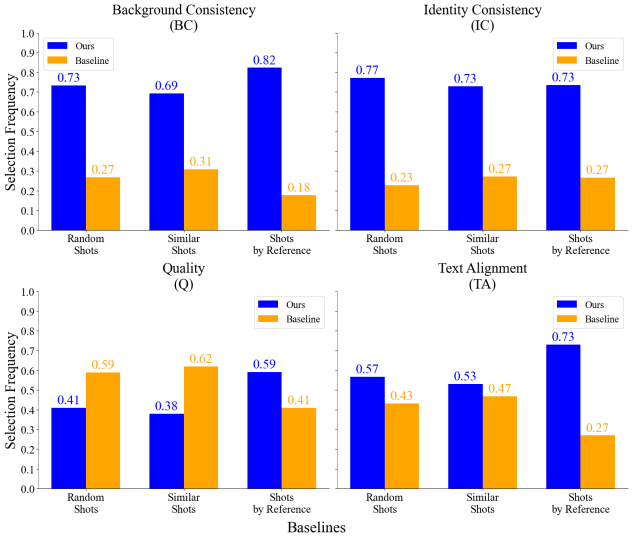

Figure 7. ***User study results.***

S.3. Motion Filtering

When curating the training dataset, we aim to select videos with significant motion. To filter such videos, we analyze three types of camera motions: pan (t_x), tilt (t_y), and zoom (s). For each video, we begin by extracting optical flow maps using RAFT [43], then estimate the homography matrices for each frame using the RANSAC algorithm. Next, we calculate the mean translation vector for the pan and tilt components. For zoom motion, we compute a divergence value that quantifies how much the pixels move towards or away from the frame center. Finally, we average these values across the video to obtain the overall motion magnitude. For t_x, t_y, and scale, we use thresholds of 8, 8, and 0.4, respectively. A video is classified as having significant motion if any of these values exceed the corresponding threshold.

S.4. Implementation Details

We employ a video diffusion model incorporating joint attention layers within its DiT blocks. The model is fine-tuned using the AdamW [26] optimizer with a learning rate of 5.0×10^{-5}, weight decay of 0.1, and betas $[0.9, 0.95]$. The learning rate scheduler follows a cosine decay strategy, with 2000 warmup steps, a decay starting at step 2000, with a minimum learning rate of 2×10^{-5}. Fine-tuning is

Figure 8. ***Additional qualitative comparison (please zoom-in)***

performed with a batch size of 128, which is 90% smaller than the size used during pretraining and runs for 5000 iterations, accounting for less than 1% of the pretraining iterations, making it a computationally lightweight approach. Note that for all baseline approaches, we use the original pre-trained checkpoint without any fine-tuning.

S.5. Additional Comparisons

Our objective—motivated by its significance in storytelling and the film industry—is to generate multi-shot videos where shots are separated by (jump) cuts, while ensuring the foreground object remains consistent, regardless of any background changes specified by the user. We perform additional comparisons with previous baselines: (1) ***SEINE*** [8] focuses on frame interpolation by generating intermediate frames between two source inputs or image-to-video generation, but does not support multi-shot text-to-video generation; (2) ***Gen-L-Video*** [48] primarily produces results for video editing that requires a source video, with 'multi-text' being used exclusively for editing purposes, restricting the model's ability to generate distinct activities across shots; (3) ***FreeNoise*** [36] and (4) ***MEVG*** [30] use multi-text to generate a continuous video and are limited in their ability to create 'jump cuts' which reduces diversity in camera angles and motion across shots. Additionally, the foreground remains fixed in the same location throughout the video and across scenes. In contrast, our approach enables greater diversity in both aspects. We compare our approach with all suggested baselines for 2 shots in Table 4. For all but FreeNoise we use the results from the respective webpages since MEVG and Gen-L-Video do not provide code for multi-text prompting, while SEINE only does frame interpolation. For FreeNoise we used our dataset and the official FreeNoise checkpoint. According to these results (in addition to the qualitative comparison in Fig. 8), we outperform all approaches across all metrics by a large margin (this also holds for >2 shots for which we do not add results here due to limited space), demonstrating our ability to generate consistent identities under multi-shot video generation settings with rich motion.

We computed the average motion in the datasets and for baselines as shown in Table 4, including continuous generation which uses the original pretrained model. We do indeed see a drop in motion (although we still have more motion than related work), and, after some investigation contribute this to our pre-processed dataset, as the average motion of our filtered dataset is reduced by 38.1% compared to the original dataset.

	MEVG	Ours	Gen-L-Video	Ours	SEINE	Ours	FreeNoise	Ours	Cont. Gen
Identity Consistency	68.5	**76.4**	66.9	**82.4**	69.1	**81.9**	68.2	**86.3**	88.9
BG Consistency	69.4	**79.8**	70.9	**85.3**	72.8	**87.4**	77.9	**89.5**	90.6
Avg. Motion	0.95	**1.36**	1.23	**1.25**	0.75	**0.97**	1.12	**1.19**	1.42

Table 4. ***Quantitative comparison with previous works***

S.6. ChatGPT Prompt Instruction

We use ChatGPT to generate our validation dataset which consists of prompts. For each prompt, we provided the instruction: "Our project involves text-to-multi-shot video generation, where each shot is controlled through local text prompts. I would like you to generate prompts for each video of N shots for 8 videos. For each shot, the background should be XX. Include one human as the foreground object and provide detailed descriptions of the human's appearance." Here, N corresponds to 2, 3, or 4, and XX specifies whether the background should be consistent or diverse. After refining the generated results, we finalized the validation prompts used in our experiments.