Versatile Transition Generation with Image-to-Video Diffusion

Supplementary Material

Outline

In this Supplementary Material, we first provide elaborated implementation details in Section 1. We also provide more details regarding our self-curated TransitBench in Section 2. Next, we incorporate the dynamic video examples into an project page and provide a detailed summary in Section 3. In Section 4, we provide more quantitative results Furthermore, we show visual comparisons of ablation studies in Section 5. We also present time complexity analysis regarding inference time in Section 6. Finally, we conclude with limitations and discussion of VTG in Section 7 and Section 8, respectively.

1. More Implementation Details

Hyper-parameters. To maximize the retention of appearance and motion priors inherent in the image-to-video diffusion backbones, we only perform the interpolated latent injection at early denoising steps (i.e., less than 5 steps). For the three interpolation parameters (i.e., λ_{noise}, λ_{lora}, and λ_{text}), we adopt linearly increasing coefficients by default. This is intuitive because the closer the generated frame is to the initial frame, the more it should resemble the content of the initial frame, and vice versa.

Inference. The inference script is built upon the codebase of DynamiCrafter [4]. To align with previous studies, we apply classifier-free guidance with a guidance scale set to 7.5. In all our experiments, the resolution and frame count of the generated videos strictly follow the benchmark specifications. For example, MorphBench requires 16 frames of 512×512 resolution video. All experiments are conducted on 1 NVIDIA A100-80GB GPU using PyTorch, with a batch size of 1. The sampling process of VTG takes $\sim$50s.

2. More details on TransitBench

TransitBench includes two subsets for concept blending and scene transition, respectively. Each subset contains 100 image pairs and each image pair constitutes the first and last frames of a transition video. All image pairs are manually collected and captioned from Google Images (under CC BY license). They are resized to 512×512. For data diversity, the categories of the image pairs in each subset are unique, ensuring that no transitions are repeated. The FPS does not affect evaluation since our model inherently supports different frame counts (e.g., 8, 32, 64) without extra tuning. We encourage readers to explore our project page, especially *Concept Blending* and *Scene Transition* sections, for a visual examination of the content and quality of TransitBench.

3. More Qualitative Results

We kindly refer readers to our project page to examine the dynamic video comparisons. We compared the visual quality of our proposed VTG with five existing methods, including DiffMorpher [5], TVG [6], SEINE [1], DynamiCrafter [4], and Generative Inbetweening [3]. For each transition task, five examples are given. We also provide two input frames along with a pair of transition captions for each example. In addition, we include several challenging cases regarding *Motion Prediction* and *Scene Transition*, which involve significant variations in motion or substantial changes in scenes between the initial and final input frames. We also summarize the successful and failed motion patterns at the bottom of the page.

4. More Quantitative Results

As shown in Table 1 and Table 2, our method outperforms other baselines in all the four metrics for both public benchmarks and TransitBench. In particular, our approach achieves significantly lower Fidelity, showcasing the better consistency between the generated frames and input frames. Note that, unlike existing methods that are specifically designed for one task (e.g., DiffMorpher for Object Morphing), VTG was designed to perform well across all the four transition tasks

5. Ablation Studies

Quantitative Ablation. As shown in Table 3, introducing LERP slightly reduced Similarity and increased Fidelity, indicating limited benefits. However, switching to SLERP significantly enhanced both Smoothness and Fidelity, showing the effectiveness of our noise initialization strategy. Adding LoRA interpolation further boosted Similarity, highlighting its capability to capture better semantics in the input frames. Incorporating BMP improved Smoothness substantially, demonstrating better motion consistency, although Alignment slightly decreased. Finally, introducing RAR achieved the best overall performance, with the highest Similarity (0.8705) and Smoothness (0.9876), while significantly reducing Fidelity (118.35). These results confirm that our method effectively enhances generation quality over a wide array of transition tasks.

Qualitative Ablation. As observed in Figure 1, introducing

Method	*Object Morphing*				*Motion Prediction*			
	Similarity (↑)	Fidelity (↓)	Smoothness (↑)	Alignment (↑)	Similarity (↑)	Fidelity (↓)	Smoothness (↑)	Alignment (↑)
DiffMorpher [5]	0.8467	54.695	0.8912	0.2535	0.7809	499.97	0.8801	0.2579
Generative Inbetweening [3]	0.6080	70.496	0.8809	0.2412	0.7573	414.83	0.8976	0.2418
TVG [6]	0.8389	50.512	0.9678	0.2708	0.8750	451.10	0.9765	0.2664
SEINE [1]	0.7389	60.828	0.9675	0.2665	0.8483	447.25	0.9620	0.2634
DynamiCrafter [4]	0.7707	54.285	0.9569	0.2538	0.8609	369.19	0.9715	0.2678
VTG (Ours)	**0.8752**	**48.071**	**0.9888**	**0.2788**	**0.8963**	**292.54**	**0.9890**	**0.2701**

Table 1. **Quantitative comparisons on object morphing and motion prediction.** We utilize two public datasets, MorphBench [5] and UCF101-7 [2], to evaluate the performance of each method.

Method	*Concept Blending*				*Scene Transition*			
	Similarity (↑)	Fidelity (↓)	Smoothness (↑)	Alignment (↑)	Similarity (↑)	Fidelity (↓)	Smoothness (↑)	Alignment (↑)
DiffMorpher [5]	0.8174	72.828	0.8954	0.2635	0.7934	86.151	0.8723	0.2624
Generative Inbetweening [3]	0.6320	86.599	0.8802	0.2428	0.7658	94.285	0.8815	0.2675
TVG [6]	0.7315	80.805	0.9605	0.2534	0.7740	80.530	0.9675	0.2726
SEINE [1]	0.7880	75.243	0.9678	0.2519	0.7954	79.833	0.9720	0.2716
DynamiCrafter [4]	0.8218	66.298	0.9738	0.2665	0.7993	74.253	0.9754	0.2745
VTG (Ours)	**0.8517**	**63.105**	**0.9845**	**0.2717**	**0.8580**	**69.689**	**0.9880**	**0.2763**

Table 2. **Quantitative comparisons on concept blending and scene transition.** Utilizing self-curated TransitBench, we evaluate the performance of each method to demonstrate their effectiveness.

Method	Similarity (↑)	Fidelity (↓)	Smoothness (↑)	Alignment (↑)
Baseline	0.8132	141.01	0.9694	0.2670
Baseline+LERP	0.7985	150.75	0.9231	0.2615
Baseline+SLERP	0.8050	125.42	0.9765	0.2669
Baseline+SLERP+LoRA	0.8601	137.92	0.9440	0.2672
Baseline+SLERP+LoRA+BMP	0.8355	160.89	0.9810	0.2527
VTG (Ours)	**0.8705**	**118.35**	**0.9876**	**0.2753**

Table 3. **Ablation study on different proposed components.** The results are averaged over four tasks. Best viewed when zoomed in.

interpolated noises to initialize the video diffusion model effectively alleviates the issue of abrupt content changes. Consequently, the intermediate generated frames become more coherent and reasonable. Furthermore, the SLERP significantly outperforms LERP, as it captures richer semantic information. Nonetheless, without LoRA interpolation, the model still struggles to capture the finest semantic details. This highlights the rationale behind our approach of applying both SLERP and LoRA interpolation on top of the baseline.

The effectiveness of our proposed frame-aware text interpolation is demonstrated in Figure 2. Given two distinct concepts, adopting text interpolation prevents abrupt content changes and avoids generating ambiguous mixed-state video frames.

Figure 3 illustrates empirical evidence for the effectiveness of our proposed bidirectional motion prediction (BMP) and representation alignment regularization (RAR). Specifically, when the start and end frames of a natural motion path are swapped, the generated transition video often exhibits a noticeable quality gap compared to the original. BMP enables the video diffusion model to accurately predict and fuse both forward and backward motion paths, allowing VTG to handle a wider range of motion prediction inputs with greater naturalness and robustness. Furthermore, incorporating RAR during fine-tuning allows the model to effectively utilize powerful representation models to capture finer semantics (e.g., a person's bangs). Another noteworthy observation is that the changes in background brightness appear more natural, contributing to smoother transitions and ultimately enhancing the fidelity of the generated transition videos.

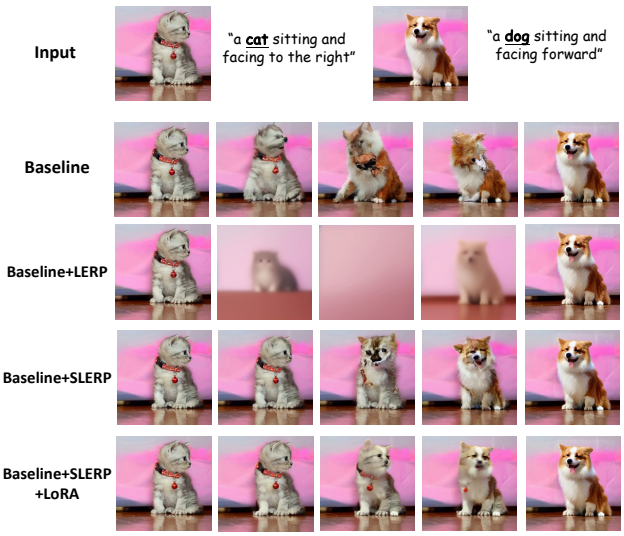

Figure 1. Qualitative results regarding the effectiveness of latent interpolation, spherical linear interpolation (SLERP) over the linear interpolation (LERP), as well as LoRA interpolation.

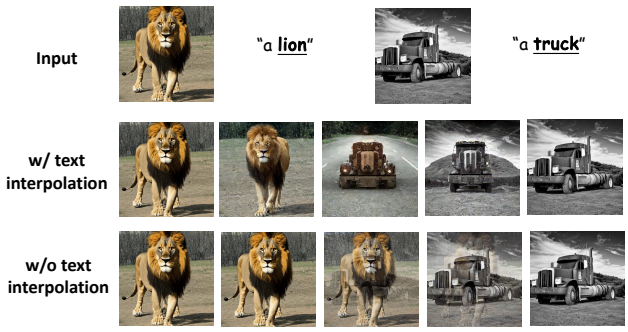

Figure 2. Qualitative results regarding the effectiveness of text interpolation.

6. Time Complexity Analysis

Method	Frame	Time (↓)
DiffMorpher [5]	16	1m18s
Generative Inbetweening [3]	16	10m17s
VTG (Ours)	16	1m25s
DiffMorpher [5]	32	2m29s
Generative Inbetweening [3]	32	19m44s
VTG (Ours)	32	2m36s

Table 4. Inference time comparison for different methods.

We compared our approach with the other two training-based methods (i.e., DiffMorpher and Generative Inbetweening). As seen in Table 4, DiffMorpher also requires LoRA training during inference, resulting in comparable runtime. In contrast, Generative Inbetweening (SVD-based) require over ten minutes for inference. To summarize, the LoRA training during VTG's inference stage does not introduce significant computational overhead, taking just over a minute for 16-frame video generation.

7. Limitations

VTG leverages existing image-to-video diffusion models as priors. It thus faces similar limitations as the diffusion backbones. One typical scenario that VTG tends to struggle is when the involved motion is very fast with complex patterns. This can be observed in Figure 4, where some intermediate frame fails to capture the moving car. We also summarize successful and failed motion patterns in our project page (last section). We will investigate how to mitigate this limitation in our future work.

8. Discussion

Adaptation to other diffusion backbones. Our proposed method can be adapted to the Stable Video Diffusion (SVD) architecture and extended for 25-frame video generation.

GPU Requirements. Our base model requires approximately 12.8 GB of GPU memory when running on a single 24 GB VRAM GPU (e.g., NVIDIA RTX 3090/4090).

More clarification on *Concept Blending* and *Scene Transition*. *Concept Blending* aims to blend two conceptually different objects for tasks like image attribute modification, data augmentation, and video generation. *Scene Transition* encompasses the evolution of the same subject across different scenes (e.g., Fig. 1-(d)), changes in the state of the same conceptual entity (e.g., Fig. 4), and the transformation of the same entity across different domains (e.g., a cartoon-style balloon transitioning into a cyberpunk-style balloon).

References

[1] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *ICLR*, 2023. 1, 2

[2] Siddhant Jain, Daniel Watson, Eric Tabellion, Aleksander Hołyński, Ben Poole, and Janne Kontkanen. Video interpolation with diffusion models. In *CVPR*, 2024. 2

[3] Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steven M Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In *ICLR*, 2025. 1, 2, 3

[4] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2024. 1, 2

[5] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Xingang Pan, and Bo Dai. Diffmorpher: Unleashing the capability of diffusion models for image morphing. In *CVPR*, 2024. 1, 2, 3

[6] Rui Zhang, Yaosen Chen, Yuegen Liu, Wei Wang, Xuming Wen, and Hongxia Wang. Tvg: A training-free transition video generation method with diffusion models. *TCSVT*, 2024. 1, 2

Figure 3. Qualitative results regarding the effectiveness of proposed bidirectional motion prediction (BMP) and representation alignment regularization (RAR).

Figure 4. **A sequence of generated intermediate frames via VTG.** Our approach tends to fail when the two input frames (i.e., the leftmost and the rightmost) involve large and complex motion.