LayerAnimate: Layer-level Control for Animation

Supplementary Material

We provide videos on the project website[1]. These videos vividly present qualitative results and a novel application of multiple layer-level control for an enhanced view experience. We recommend that readers watch these videos, as they provide a clearer and more intuitive understanding of this paper.

A. Motion-based Hierarchical Merging

We showcase the illustration of *Motion-based Hierarchical Merging (MHM)* in Fig. S1. MHM regards masklets as nodes and constructs a treemap using hierarchical clustering based on motion scores, merging layers with similar motion scores from the bottom up. Considering the variability in layer numbers during production, we do not restrict a fixed number of layers. Instead, we define the maximum layer capacity N, which is much less than the number of masklets, and a motion score merging threshold η_s. Layers are merged from the bottom up until the count of layers falls below the capacity N and the motion score difference exceeds the threshold η_s.

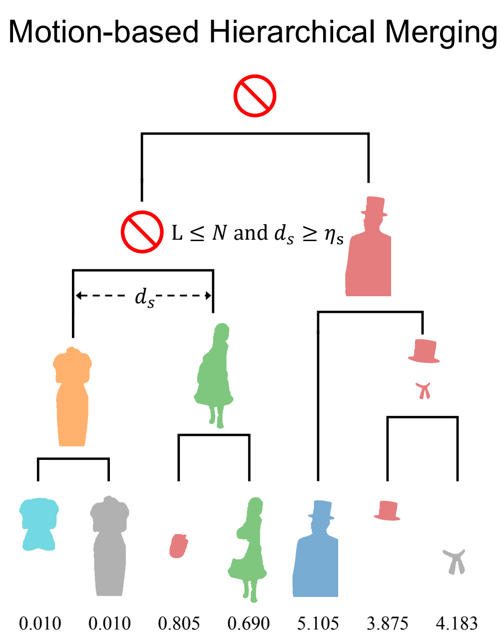

Figure S1. **Motion-based Hierarchical Merging.** Layers are merged from the bottom up until the layer count L falls below the maximum layer capacity N and the motion score difference d_s exceeds the threshold η_s.

[1] https://layeranimate.github.io

Method	Task	**MSE↓**
DragAnything	I2V	1398.39
Tora	I2V	352.43
LayerAnimate (offset)	I2V	50.02
LayerAnimate (heatmap)	I2V	47.23
LayerAnimate (ours)	I2V	**46.58**
Framer	Interpolation	44.91
LayerAnimate (ours)	Interpolation	**24.40**

Table S1. **Comparison of trajectory control performance.**

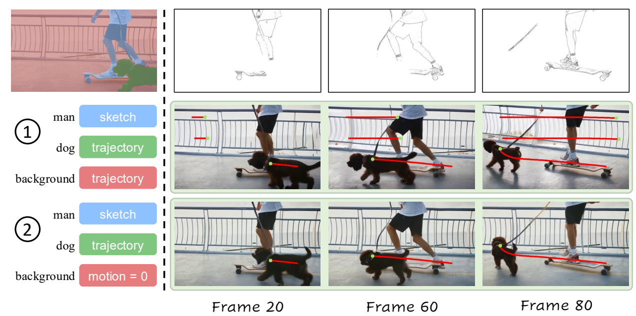

Figure S2. **Real-world cases.** LayerAnimate supports composite control in real world such as skateboarding with dynamic/static backgrounds.

B. Trajectory Control

To fully elucidate the performance of trajectory control, we evaluate the Mean Squared Error (MSE) between the predicted animations and the ground truth object trajectories in Tab. S1.

C. Application to Real-World Domains

While our framework, LayerAnimate, is primarily designed and evaluated for cartoon animation, its core principle of layer-level control possesses significant potential for extension into real-world video generation.

We showcase this potential in Fig. S2, where LayerAnimate supports composite control in real-world scenarios, such as skateboarding with dynamic and static backgrounds. This demonstrates the framework's capability to handle complex textures and natural motion, opening avenues for applications in video editing and special effects.

Furthermore, to enhance the applicability and performance of our framework on high-resolution content, we have integrated a more advanced base model, Wan2.1. This integration elevates the generation capabilities of LayerAnimate, enabling animation generation up to **81 frames at** 480×832 **resolution**. The enhanced model and associated

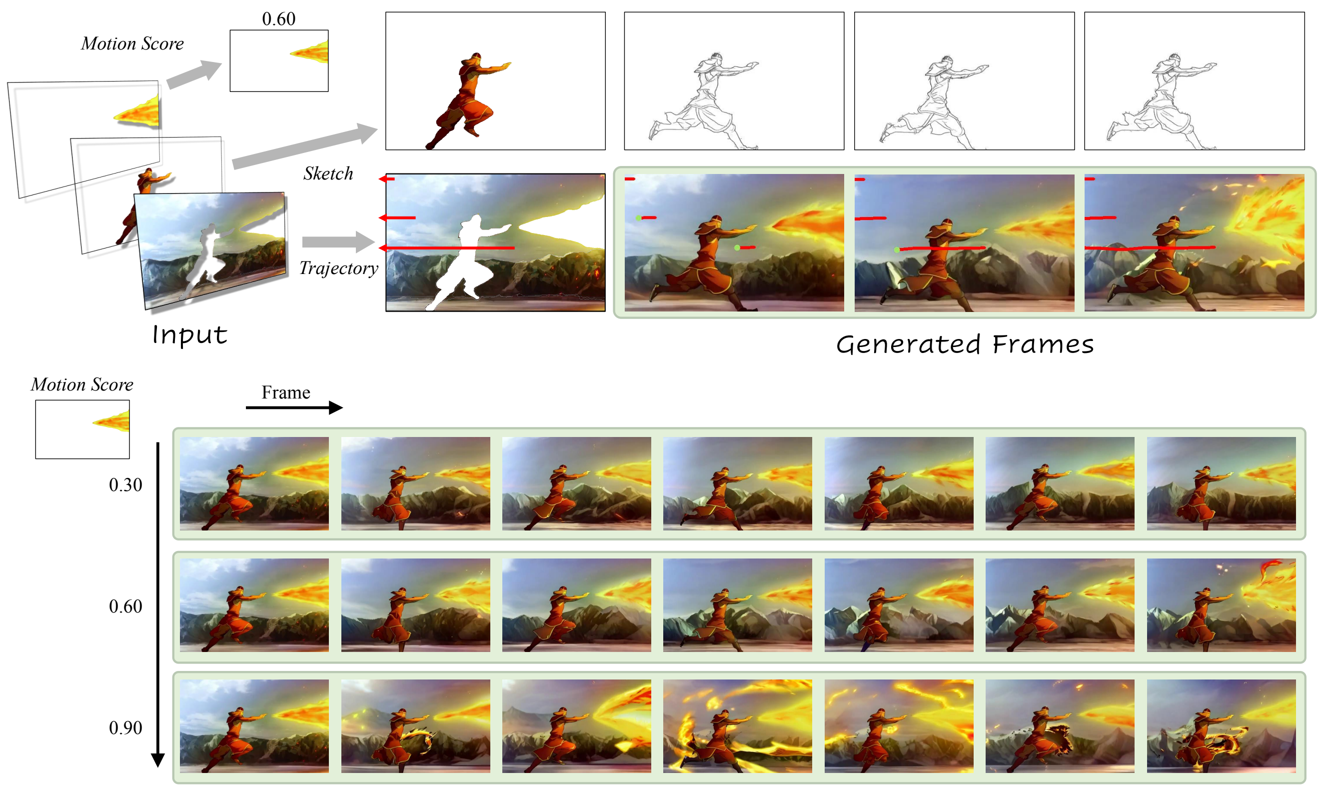

Figure S3. **Impact of different motion scores.**

code have been open-sourced, with the goal of fostering further research and development in controllable video generation.

D. Analysis of Layer Capacity

A key design choice in our framework is the number of layers used to represent a scene. In practice, we find that four layers typically suffice to represent the essential compositional elements in most cartoon scenes, such as the background, primary characters, and key foreground objects. We conduct experiments with more layers (Tab. S2) and observe marginal improvements but provide finer-grained control. However, this increased granularity comes at the cost of greater annotation effort and a more complex user interface. Therefore, we determine that four layers is an optimal balance between expressive control and user-friendliness, enabling multi-layer manipulation while offering an easy-to-use interface.

I2V w/ score	**N = 1**	**N = 2**	**N = 4**	**N = 6**	**N = 8**
FVD↓	87.88	81.93	**81.36**	82.23	82.29
FID↓	14.63	14.15	13.84	**13.82**	13.91

Table S2. **Ablation study on layer capacity.**

E. Motion Score

Here, we vividly illustrate the impact of adjusting the motion score on generation in Fig. S3 using the sample from the teaser figure.

F. More Qualitative Results

To further showcase the versatility and practical utility of our framework, we present a range of additional applications in Fig. S4. These examples move beyond simple motion generation to highlight the innovative and user-friendly control that LayerAnimate offers for complex animation tasks. These results underscore the creative freedom and precise control that LayerAnimate provides, making it a powerful tool for both novice and expert animators.

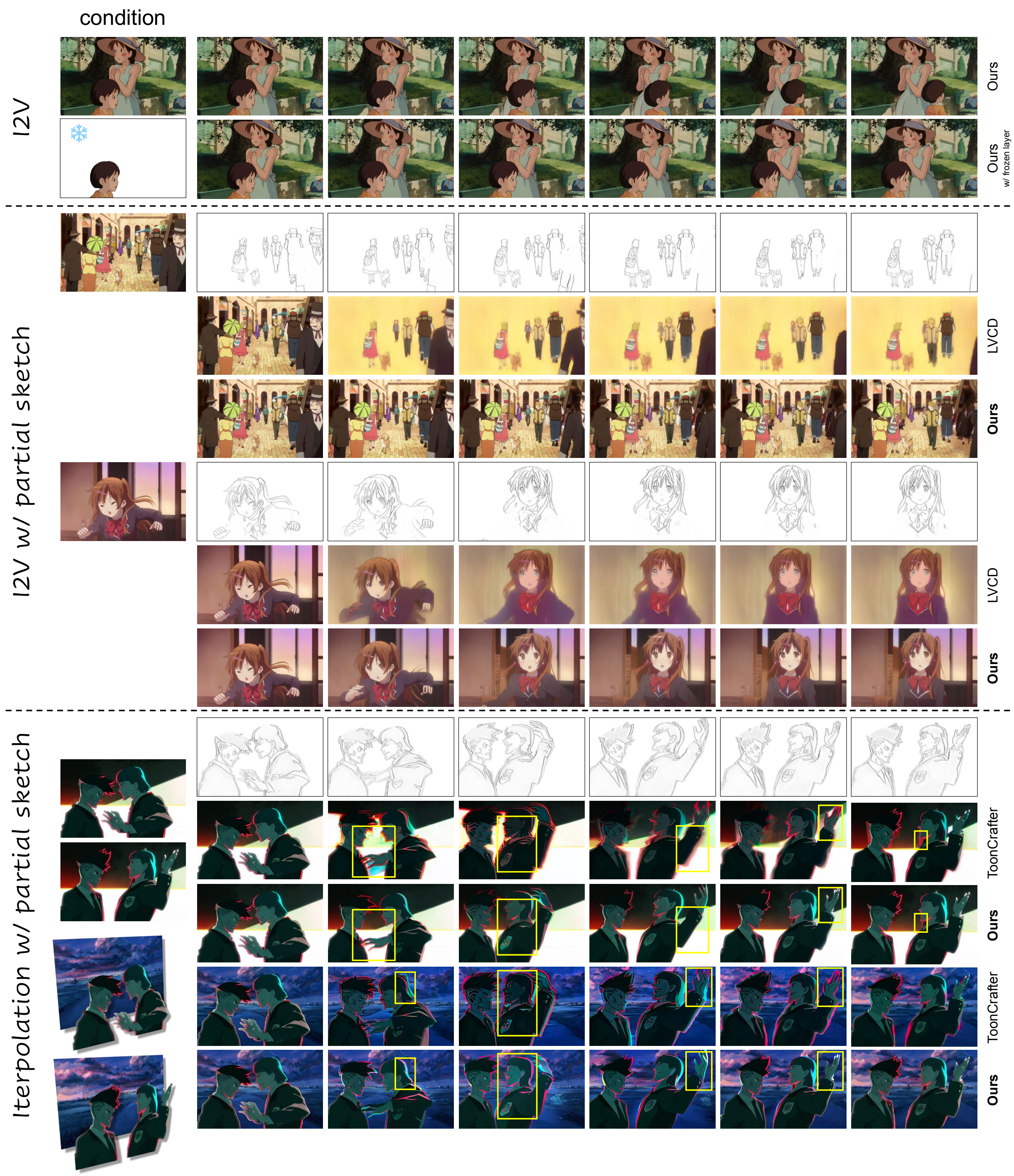

Figure S4. **Layer-level application.** LayerAnimate provides innovative and user-friendly control options for animation, enabling users to freeze specific elements, animate characters with partial sketches, and switch dynamic backgrounds. The layer-level control over individual layers ensures that foreground layers remain consistent and nearly unaffected by background changes.