PhysAnimator: Physics-Guided Generative Cartoon Animation

Supplementary Material

1. Motion Equation

To model the fluidity and exaggerated motions typical of anime, we represent anime objects as deformable bodies and adopt the Fixed Corotated model [5] as the constitutive model. The dynamics of the deformable bodies are governed by Newton's Second Law, expressed as:

$$\frac{d^2\boldsymbol{x}}{dt^2} = \boldsymbol{M}^{-1}(\boldsymbol{f}_{\text{int}}(\boldsymbol{x}) + \boldsymbol{f}_{\text{ext}}(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{M}$ is the mass matrix, representing the masses of all vertices:

$$\boldsymbol{M} = \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & m_N \end{pmatrix}. \tag{2}$$

The mass of i-th vertex is calculated as

$$m_i = \sum_{\mathcal{T}_j \in S_i} \frac{\rho_j V_j}{3} \tag{3}$$

where S_i is the set of triangles containing the i-th vertices. ρ_j and V_j denote the mass density and volume of j-th triangle, respectively. The internal force fir the $\boldsymbol{f}_{\text{int}}(\boldsymbol{x})$ of Fixed Corotated deformable model can be derived as

$$\begin{aligned} \boldsymbol{f}_{\text{int}}(\boldsymbol{x}) &= -\frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}} \\ &= -\frac{\partial \sum_{i=1}^{N} \Psi(\boldsymbol{F}_i) V_i}{\partial \boldsymbol{x}} \\ &= -\sum_{i=1}^{N} \left(\frac{\partial \Psi(\boldsymbol{F}_i)}{\partial \boldsymbol{F}_i} \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{x}} V_i \right), \end{aligned} \tag{4}$$

where

$$\begin{aligned} \frac{\partial \Psi(\boldsymbol{F}_i)}{\partial \boldsymbol{F}_i} &= \frac{\partial (\mu \|\boldsymbol{F}_i - \boldsymbol{R}_i\|_F^2 + \frac{\lambda}{2}(\det(\boldsymbol{F}_i) - 1)^2)}{\partial \boldsymbol{F}_i} \\ &= 2\mu(\boldsymbol{F}_i - \boldsymbol{R}) + \lambda\,(\det(\boldsymbol{F}_i) - 1)) \det(\boldsymbol{F}_i) \boldsymbol{F}_i^{-T}, \end{aligned} \tag{5}$$

The deformation gradient $\boldsymbol{F}_i$ for a 2D triangle $\mathcal{T}_i$ is computed as:

$$\boldsymbol{F}_i = [\boldsymbol{x}_1 - \boldsymbol{x}_0, \boldsymbol{x}_2 - \boldsymbol{x}_0][\boldsymbol{X}_1 - \boldsymbol{X}_0, \boldsymbol{X}_2 - \boldsymbol{X}_0]^{-1}. \tag{6}$$

Here $\{\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2\}$ and $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2\}$ represent the positions of vertices of triangle $\mathcal{T}_i$ in the undeformed (material) space and the deformed (world) space respectively.

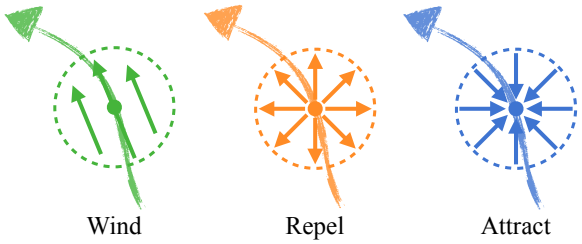

Figure 1. **Flow Particles**. Wind particles can simulate wind effects, creating natural swaying motions; repel particles act as collision barriers, adjusting object trajectories; attract particles can pull objects toward specific points as desired.

To solve the motion equation (1), we apply a semi-implicit Euler method, discretizing it as:

$$\begin{aligned} \boldsymbol{x}^{n+1} &= \boldsymbol{x}^n + \Delta t \boldsymbol{v}^{n+1}, \\ \boldsymbol{v}^{n+1} &= \boldsymbol{v}^n + \Delta t \boldsymbol{M}^{-1}(\boldsymbol{f}_{\text{int}}(\boldsymbol{x}^n) + \boldsymbol{f}_{\text{ext}}(\boldsymbol{x}^n)), \end{aligned} \tag{7}$$

where $\boldsymbol{x}^n, \boldsymbol{v}^n$ and $\boldsymbol{x}^{n+1}, \boldsymbol{v}^{n+1}$ denote the positions and velocities at time steps t^n and t^{n+1}, respectively. The timestep size Δt is set to 0.001 in our experiments. By iteratively applying Eq. 7 from $t = 0$, we obtain the physical states (positions and velocities) at any future time step.

2. Details of Interactive Animation

To enable interactive animation, we introduce energy strokes that carry flow particles, which exert forces on nearby vertices to drive motion. As illustrated in Fig. 1, we provide three types of customizable energy strokes: wind, repel, and attract.

For a wind flow particle located at position $\boldsymbol{p}$ and a nearby vertex at position $\boldsymbol{q}$, the exerted force is defined as:

$$\begin{aligned} \boldsymbol{f}_{\text{wind}}(\boldsymbol{p}, \boldsymbol{q}) &= s(1 - w(\boldsymbol{p}, \boldsymbol{q}))\boldsymbol{d}, \\ w(\boldsymbol{p}, \boldsymbol{q}) &= \frac{\|\boldsymbol{p} - \boldsymbol{q}\|}{r}, \end{aligned} \tag{8}$$

where r is the influence range, s represents the particle's strength, and $\boldsymbol{d}$ denotes the particle's movement direction. Similarly, the forces exerted by repel and attract flow particles are given as:

$$\begin{aligned} \boldsymbol{f}_{\text{repel}} &= s\frac{\boldsymbol{q} - \boldsymbol{p}}{r}, \\ \boldsymbol{f}_{\text{attract}} &= s\left(1 - \frac{\|\boldsymbol{q} - \boldsymbol{p}\|}{r}\right)\frac{\boldsymbol{p} - \boldsymbol{q}}{\|\boldsymbol{p} - \boldsymbol{q}\|}. \end{aligned} \tag{9}$$

In addition to flow particles, we incorporate rigging point support to allow animators to fix specific regions or define

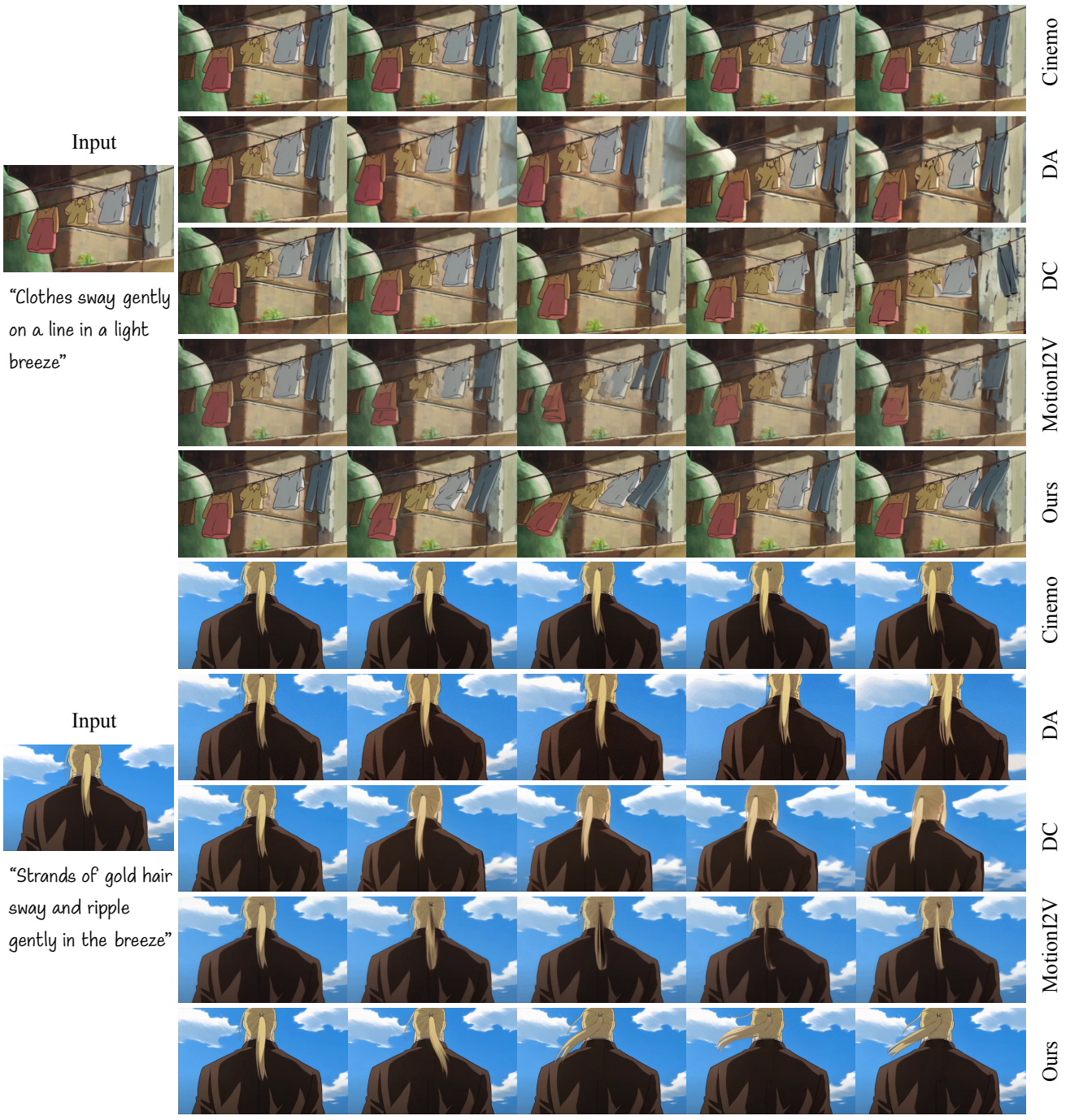

Figure 2. **Additional Qualitative Comparison**. We provide additional comparison results against Cinemo [3], Drag Anything (DA) [6], DynamiCrafter (DC) [8] and Motion-I2V [4].

desired motion trajectories. Inspired by the swaying effects commonly seen in anime, we design a periodic wavy rigging point. The position of a wavy rigging point at time t is expressed as:

$$\boldsymbol{x}_r(t) = \boldsymbol{X}_r + s\sin(ft)\boldsymbol{d}_r, \tag{10}$$

where $\boldsymbol{X}_r$ is the position in the undeformed state, f is the swaying frequency, and $\boldsymbol{d}_r$ represents the swaying direction. The selected region will follow the rigging point's motion, creating desired animation effects.

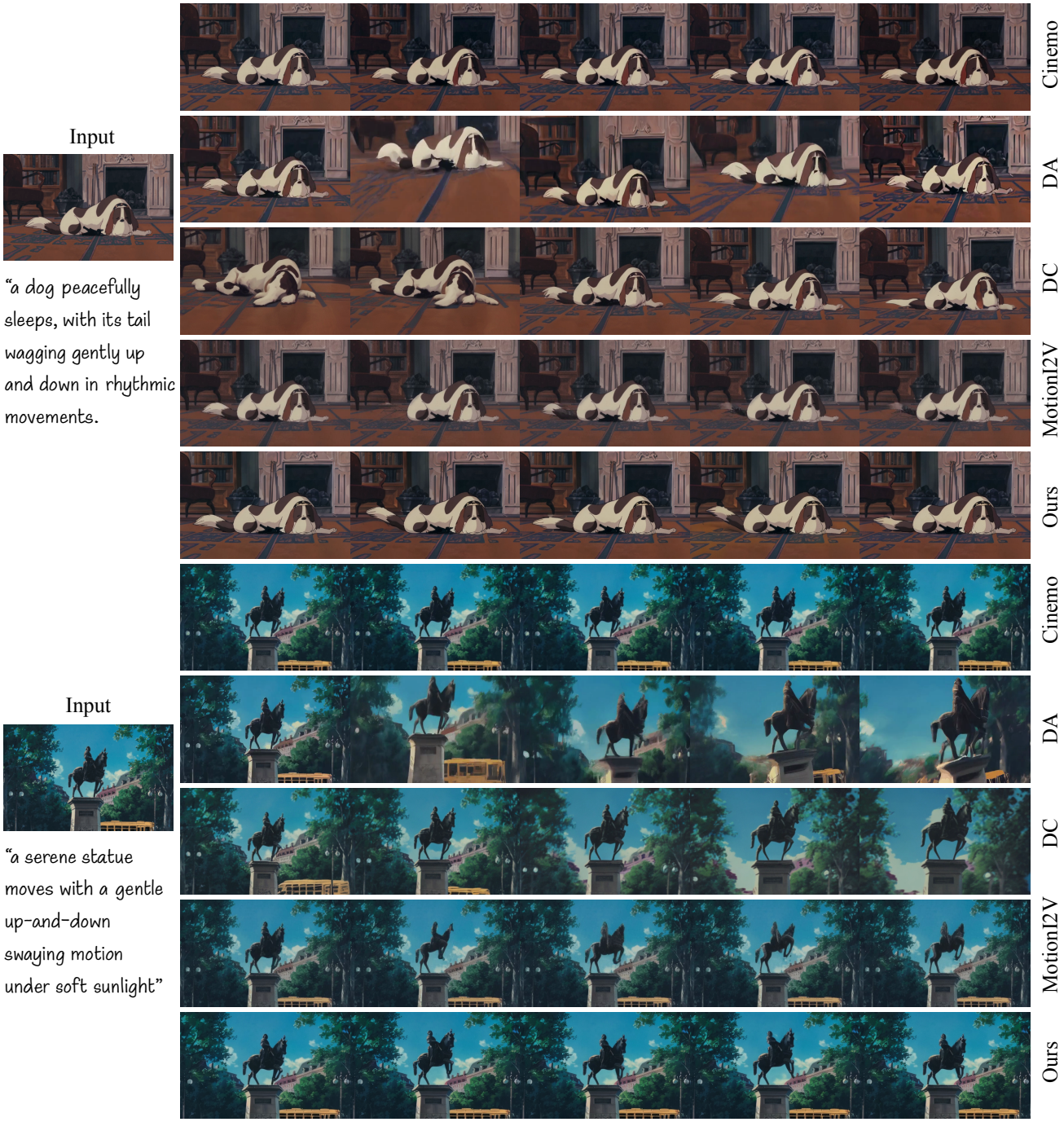

Figure 3. **Additional Qualitative Comparison**. We provide additional comparison results against Cinemo [3], Drag Anything (DA) [6], DynamiCrafter (DC) [8] and Motion-I2V [4].

3. Training Details

We adopt the standard ControlNet [9] design, integrating each frame control signal into the ToonCrafter [7]. For LVCD [2], we use the same network architecture but pre-process the input sketch with Gaussian blur. Both are trained on 8 A100 GPUs with the learning rate 1e-5 and batch size 32. The ControlNet in ToonCrafter is trained from the SD2.1 model for 50K steps. LVCD is fine-tuned from its pretrained for 10K steps. We will carefully add these details to our revised appendix.

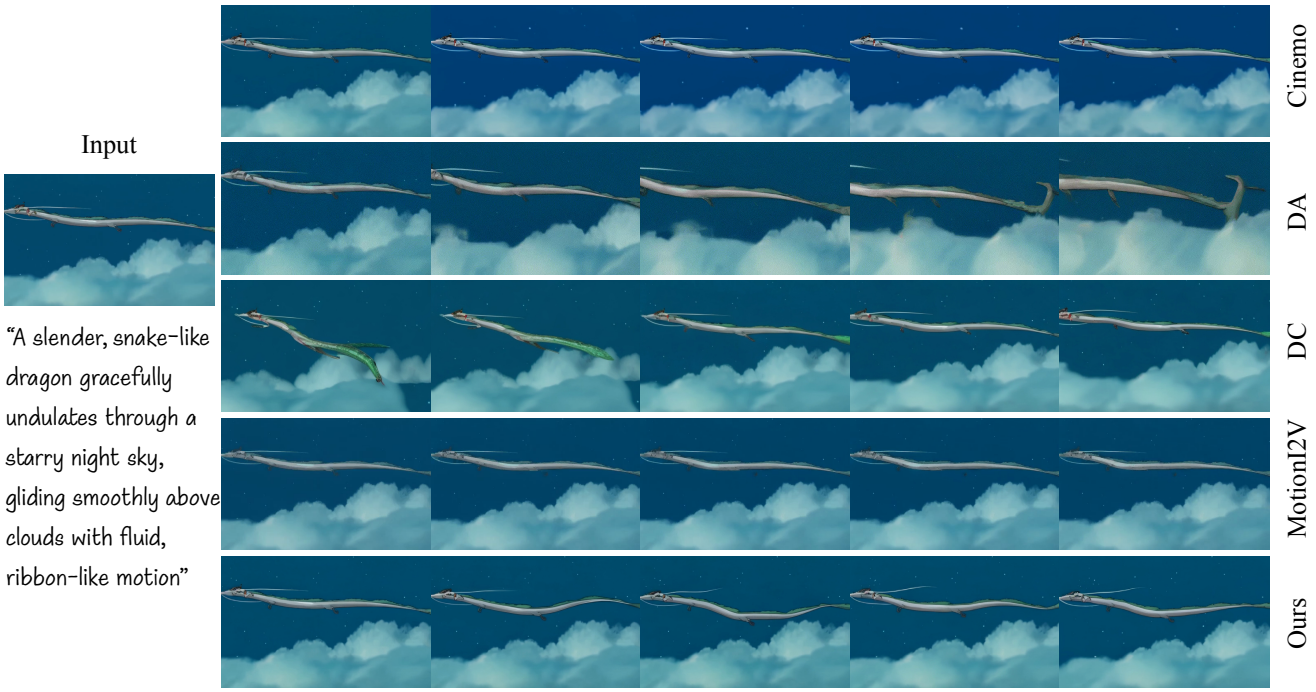

Figure 4. **Additional Qualitative Comparison**. We provide additional comparison results against Cinemo [3], Drag Anything (DA) [6], DynamiCrafter (DC) [8] and Motion-I2V [4].

Table 1. **Ablation of Dynamics Enhancement.** We present the evaluation results for our Dynamics Enhancement (DE) module. The metrics VSVQ, VSTC, VSDD, and VSFC represent VideoScore [1] assessments of visual quality, temporal consistency, dynamic degree, and factual consistency, respectively. User preferences are determined based on which video resembles real anime more.

Methods	User Pref.↑	VSVQ↑	VSTC↑	VSDD↑	VSFC↑
Ours w/o DE	29.6%	2.94	2.90	2.48	2.68
Ours w/ DE	70.4%	2.89	2.86	2.48	2.64

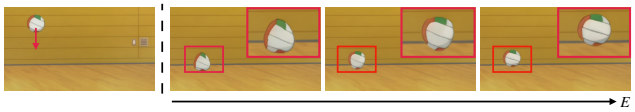

Figure 5. **Controllable Generation with Material Property.** As the Young's modulus E of the ball increases from left to right, the ball becomes more capable of maintaining its original shape under external forces.

4. More Results

4.1. Controllable Generation

Our physics-based modeling approach offers flexibility in adjusting the properties of animated objects, such as stiffness, as shown in Fig. 5. Additionally, Fig.6 demonstrates how our method enables the convenient creation of diverse dynamic effects by applying energy brushes with varying settings.

Figure 6. **Controllable Generation with Energy Stroke.** By applying energy strokes with varying directions and strengths, our method enables the convenient generation of diverse dynamic effects.

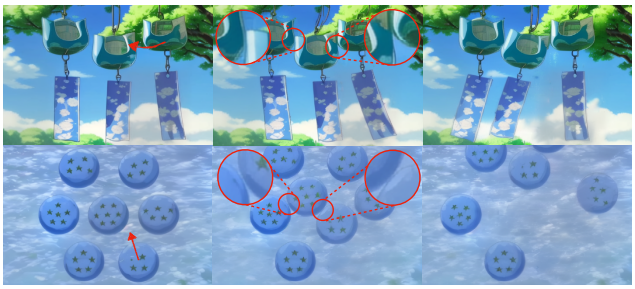

Figure 7. **Multi-Object Interaction.** Collisions are highlighted with red circles.

4.2. Multi-object Interaction

Our approach inherently supports multi-object interactions, provided with simulated inter-object interactions, such as collision handling, shown in Figure 7.

5. Ablation Study

We perform an ablation study to evaluate the effectiveness of our complementary dynamics enhancement module. As part of this evaluation, we conducted a user study where participants were asked, "Which video resembles real anime more?" We present the user preferences alongside the VideoScore [1] metrics in Table 1. Although applying the dynamics enhancement results in a slight decrease in VideoScore metrics, the user study shows a clear preference (70.4%) for the videos with dynamics enhancement, demonstrating this module's contribution to improved anime stylization.

6. More Comparison Results

In Fig. 2, 3 and 4, we provide additional qualitative comparison results with baseline methods, including Cinemo [3], Drag Anything [6], DynamiCrafter [8] and Motion-I2V [4].

References

[1] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. *arXiv:2406.15252*, 2024. 4, 5

[2] Zhitong Huang, Mohan Zhang, and Jing Liao. Lvcd: Reference-based lineart video colorization with diffusion models. *arXiv:2409.12960*, 2024. 3

[3] Xin Ma, Yaohui Wang, Gengyu Jia, Xinyuan Chen, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Cinemo: Consistent and controllable image animation with motion diffusion models. *arXiv:2407.15642*, 2024. 2, 3, 4, 5

[4] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3, 4, 5

[5] Alexey Stomakhin, Russell Howes, Craig A Schroeder, and Joseph M Teran. Energetically consistent invertible elasticity. In *Symposium on Computer Animation*, 2012. 1

[6] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 2, 3, 4, 5

[7] Jinbo Xing, Hanyuan Liu, Menghan Xia, Yong Zhang, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Tooncrafter: Generative cartoon interpolation. *arXiv:2405.17933*, 2024. 3

[8] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2025. 2, 3, 4, 5

[9] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3