ART·V: Auto-Regressive Text-to-Video Generation with Diffusion Models Supplementary Material

Wenming Weng[1] *, Ruoyu Feng[1], Yanhui Wang[1], Qi Dai[2], Chunyu Wang[2], Dacheng Yin[1], Zhiyuan Zhao[2], Kai Qiu[2], Jianmin Bao[2], Yuhui Yuan[2], Chong Luo[2] †, Yueyi Zhang[1], Zhiwei Xiong[1]
[1]University of Science and Technology of China [2]Microsoft Research Asia
https://warranweng.github.io/art.v

1. Implementation Details

We add more implementation details of network architecture, training and inference in this section.

ART·V Architecture. ART·V is composed of two individual networks, *i.e.*, mask prediction network Φ_{mask} and dynamic noise prediction network $\Phi_{dynamic}$, for estimating mask and dynamic noise, respectively. Both networks utilize the same architecture except for the minor modifications of feature channel number. We report the architecture details in Tab. 1. As can be seen, we reduce the feature channel of Φ_{mask} compared with $\Phi_{dynamic}$. The parameter of Φ_{mask} is 51.18 M, which is much smaller than $\Phi_{dynamic}$ of 1167.69 M. Because we utilize Φ_{mask} to predict the one-channel mask, which is easier compared with dynamic noise estimation of $\Phi_{dynamic}$. The autoencoder and text encoder of ART·V are elaborated in Tab. 3. We use AutoencoderKL [8] and FrozenOpenCLIPEmbedder [7] to initialize the autoencoder and text encoder of ART·V. We adopt the default settings of T2I-Adapter [5] except for channel settings. Please check the adapter setting in [3].

Training. We report the training details in Tab. 4. We follow most default training settings as in [8] to train ART·V. Thanks to the 2D architecture of ART·V, we can use a large batch size of 480 to conduct end-end training with the limited GPU resources.

Inference. We use DPMPP2SAncestral Sampler [1] to conduct inference. In order to save inference time, the sampling step is set as 50. We found that increasing sampling step can not bring a notable quality boot. We choose 50 to make a good speed-quality trade-off. To amplify the effect of the conditional signals of reference frames $\boldsymbol{y}_{ref}$, global anchor frame $\boldsymbol{y}_{anchor}$ and text prompts $\boldsymbol{y}_{text}$, we adopt the classifier-free guidance [4] for inference. In specific, the final predicted noise can be formulated as

$$\begin{aligned}\epsilon &= \Phi(\boldsymbol{y}_{ref}, \boldsymbol{y}_{anchor}, \boldsymbol{y}_{text}) \\ &+ \omega_{ref}(\Phi(\boldsymbol{y}_{ref}, \boldsymbol{y}_{anchor}, \boldsymbol{y}_{text}) - \Phi(\emptyset, \boldsymbol{y}_{anchor}, \boldsymbol{y}_{text})) \\ &+ \omega_{anc}(\Phi(\boldsymbol{y}_{ref}, \boldsymbol{y}_{anchor}, \boldsymbol{y}_{text}) - \Phi(\boldsymbol{y}_{ref}, \emptyset, \boldsymbol{y}_{text})) \\ &+ \omega_{txt}(\Phi(\boldsymbol{y}_{ref}, \boldsymbol{y}_{anchor}, \boldsymbol{y}_{text}) - \Phi(\boldsymbol{y}_{ref}, \boldsymbol{y}_{anchor}, \emptyset)),\end{aligned} \tag{1}$$

where ω_{ref}, ω_{anc} and ω_{txt} are the guidance scales of $\boldsymbol{y}_{ref}$, $\boldsymbol{y}_{anchor}$ and $\boldsymbol{y}_{text}$. We set ω_{ref}, ω_{anc} and ω_{txt} as 0.25, 0.25 and 6.5, respectively. The values may be changed for different samples to achieve better quality.

Table 1. Network architecture details. We initialize $\Phi_{dynamic}$ using the pretrained SD-2.1 [8]. Φ_{mask} is randomly initialized.

Setting	$\Phi_{dynamic}$	Φ_{mask}
input_shape	[4, 80, 80]	[4, 80, 80]
output_shape	[4, 80, 80]	[1, 80, 80]
model_channels	320	64
attention_resolutions	[4, 2, 1]	[4, 2, 1]
num_res_blocks	2	2
channel_mult	[1, 2, 4, 4]	[1, 2, 4, 4]
num_head_channels	64	32
transformer_depth	1	1
context_dim	1024	1024
adapter_config:		
channels	[320, 640, 1280, 1280]	[64, 128, 256, 256]
nums_rb	2	2
cin	8	8
ksize	1	1
sk	True	True
use_conv	False	False
params (M)	1167.69	51.18

* This work is done when the author is an intern with MSRA.
† Corresponding author.
[1] https://github.com/Stability-AI/generative-models/blob/main/sgm/modules/diffusionmodules/sampling.py#L247

Table 2. Model efficiency comparisons. A batch is a video containing 16 frames. We choose three resolution settings of 320, 448 and 768 for inference. All experiments are conducted in one Nvidia A100-80GB GPU.

Method	Inference									Training			
	FLOPs (G/batch)			Throughput (batch/s)			GPU memory (GB/forward)			Batch size	Iteration (k)	GPU number	GPU type
	320	448	768	320	448	768	320	448	768				
ModelScope [9]	3689.72	7201.22	21100.92	7.87	3.43	0.78	10.91	16.67	75.08	3200	267	-	A100-80GB
ART•V (Ours)	3163.20	6162.88	18036.96	12.16	5.55	1.35	10.52	11.08	13.44	480	258	4	A100-80GB

Table 3. Details of autoencoder and text encoder.

Setting	Value
	Autoencoder
type	AutoencoderKL [8]
z_channels	4
in_channels	3
out_ch	3
ch	128
ch_mult	[1, 2, 4, 4]
num_res_blocks	2
	Textencoder
type	FrozenOpenCLIPEmbedder [7]
Embedding dimension	1024
CA resolutions	[1, 2, 4]
CA sequence length	77

Table 4. Training details.

Setting	Value
Diffusion config:	
loss	mean squared error
timesteps	1000
noise schedule	linear
linear start	0.00085
linear end	0.0120
prediction model	eps-pred
optimizer	AdamW
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
learning rate	$1e^{-5}$
batch size	480
EMA decay	0.9999
GPU num	4
Training data FPS	8

2. Model Efficiency Evaluation

We evaluate the model efficiency of our ART•V and ModelScope [9] in this section. All experiments are conducted in one Nvidia A100-80GB GPU. Notably, ModelScope generates a whole video from text in a one-shot manner. We compare the statistics for generating short video clips containing 16 frames. Table 2 shows the results. ART•V requires slightly fewer FLOPs than ModelScope, while enjoying an almost 2× faster inference speed. The GPU memory cost of ART•V is much lower than that of ModelScope when performing inference at high resolution. In addition, the training cost of ART•V is significantly reduced compared to ModelScope, where the latter demands hundreds of GPUs to allow training on large batch size of 3200.

3. Investigation of Masked Diffusion Model

In this section, we provide more visualizations of mask predicted by masked diffusion model. The reference frame is generated by SDXL [6]. The maximal sampling step is 50. We visualize the mask with an interval of 4. The results are presented in Fig. 1. It is worth noting that the black area of the mask has the low value, which means motion area that needs to be predicted by dynamic noise prediction network. In contrast, the bright area of the mask has the high value, which can be directly copied from the reference frame.

4. Additional Experiments

We provide more visual results of text-to-video generation in Fig. 7, Fig. 8, Fig. 9, Fig. 10, and more visual results of text-image-to-video generation in Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6, and more visual results of multi-prompt text-to-video generation in Fig. 11.

For text-to-video generation, we make comparisons with one well-known method ModelScope [9]. In particular, our ART•V, specifically trained for text-image-to-video generation, can generate comparable and even better results in comparison with ModelScope [9].

For text-image-to-video generation, we make comparisons with one powerful image-to-video method I2VGen-XL [1]. We use Midjourney [2] to generate the initial frame. As can be clearly observed, I2VGen-XL [1] can not keep the original details of reference frame. It only captures the conceptual style and generate very limited and unrealistic motions. In contrast, our ART•V captures large motion while preserves the overall scene, showcasing rich details and maintaining aesthetic quality.

For multi-prompt text-to-video generation, we collect multiple prompts, each representing specified scene and motion. We generate 16 frames for each prompt. We fix the global anchor frame for each prompt in order to keep scene consistency. In specific, for the first prompt, we choose the first generated frame by the model as the global anchor frame. For the following prompts, the global anchor frame is initialized from the last generated frame of 16 frames of previous consecutive adjacent prompt.

We also provide an additional video demonstration in the supplementary video. We recommend referring to it for the qualitative comparison.

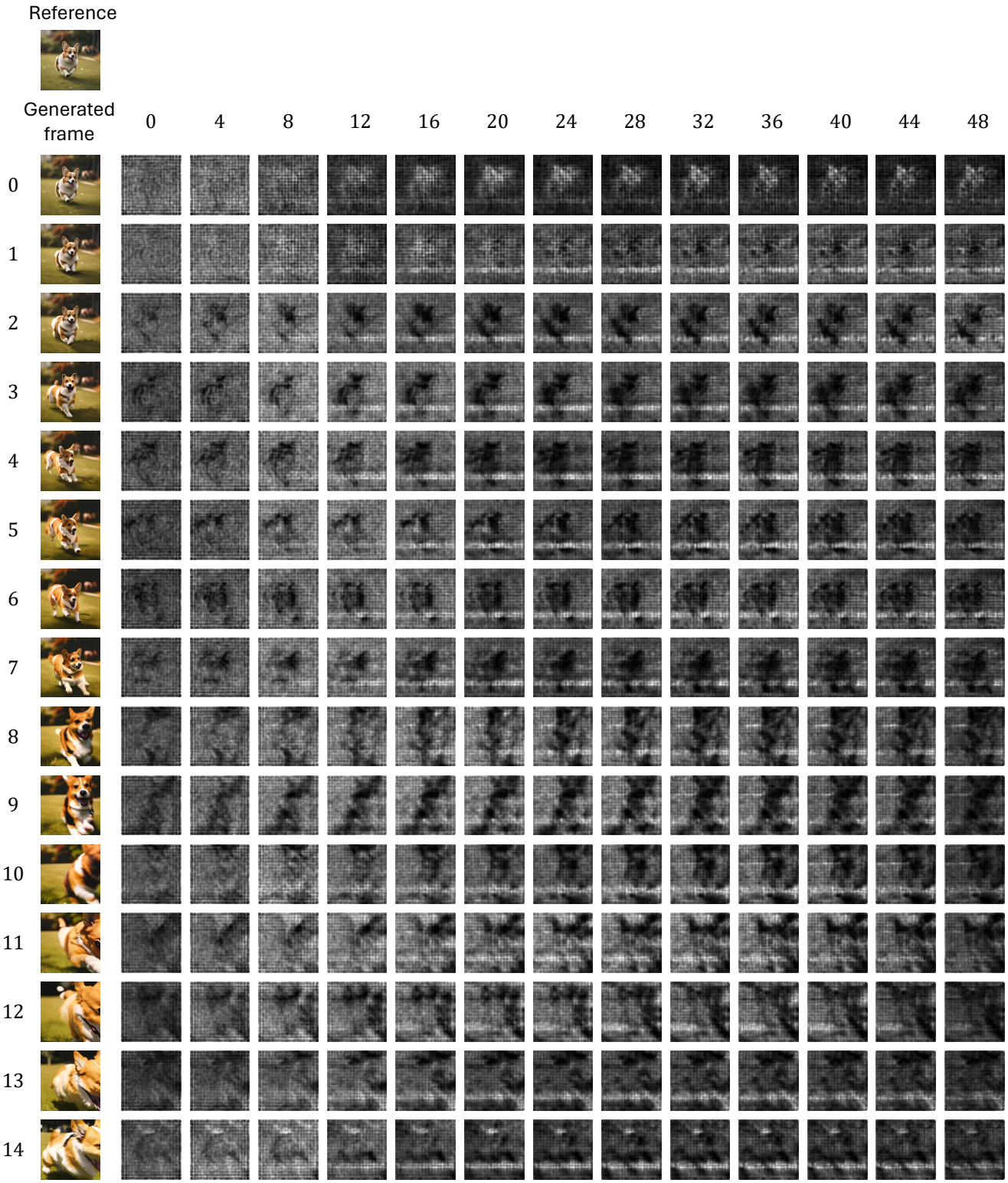

Figure 1. Visualization of mask predicted by masked diffusion model.

Figure 2. Visual results of text-image-to-video generation.

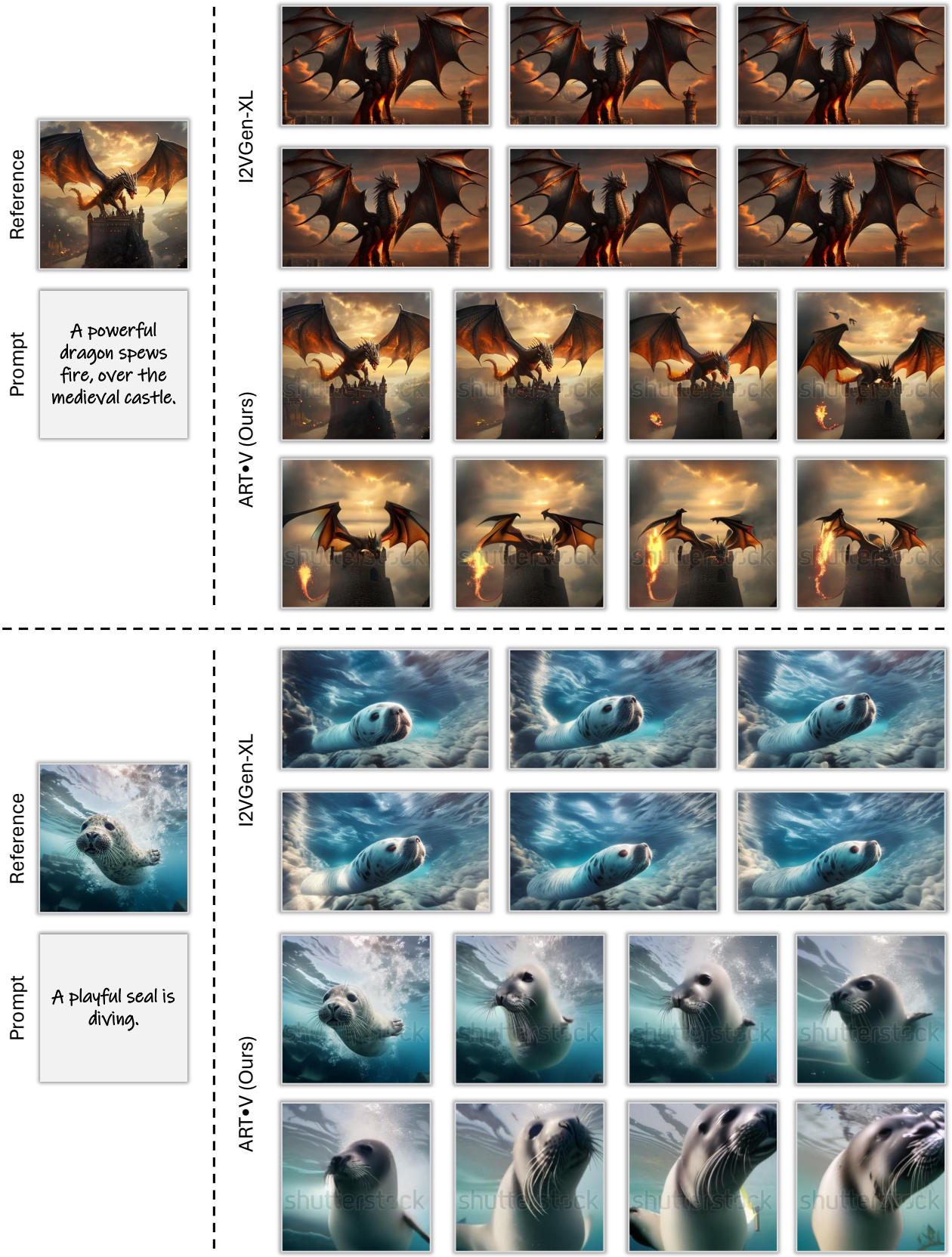

Figure 3. Visual results of text-image-to-video generation.

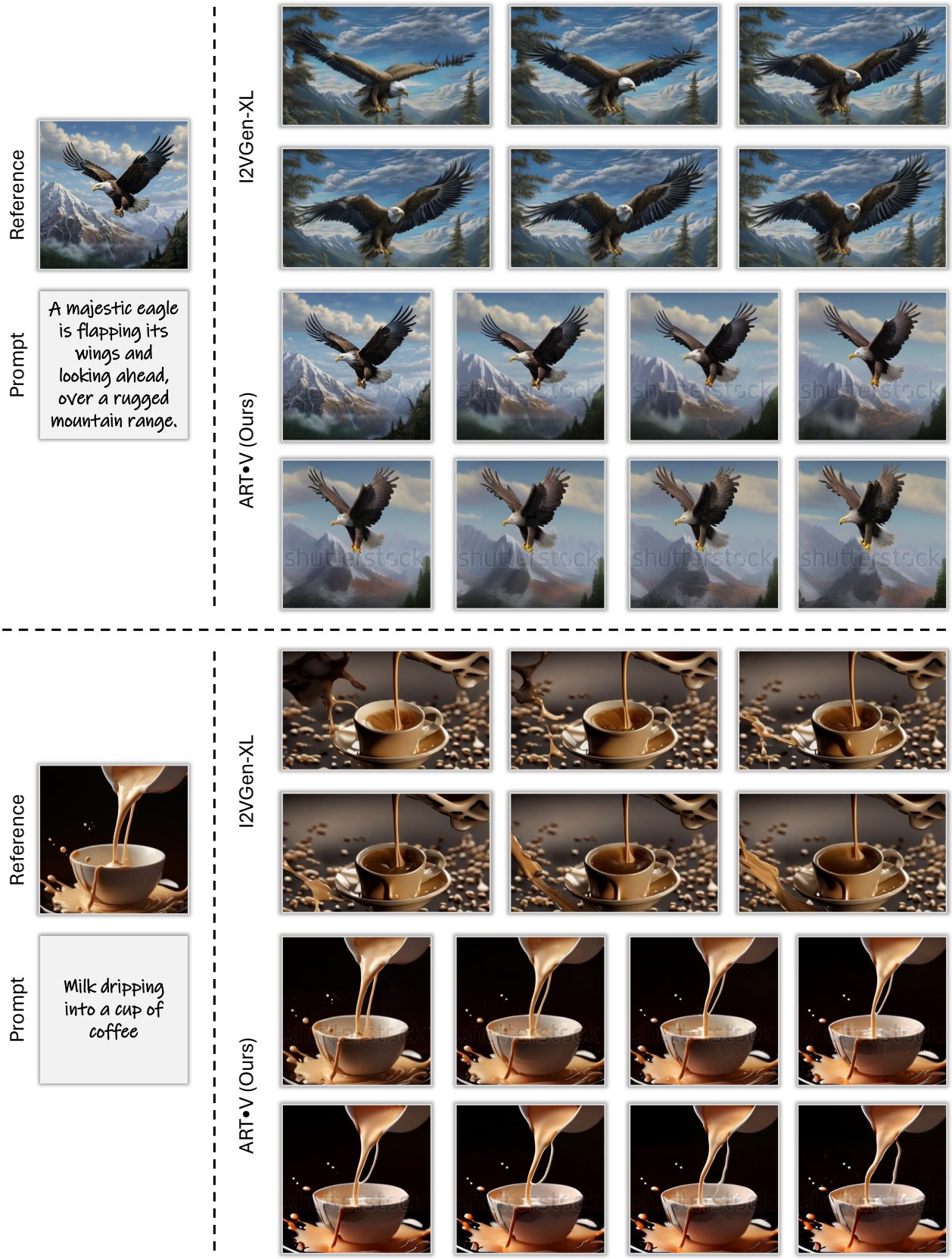

Figure 4. Visual results of text-image-to-video generation.

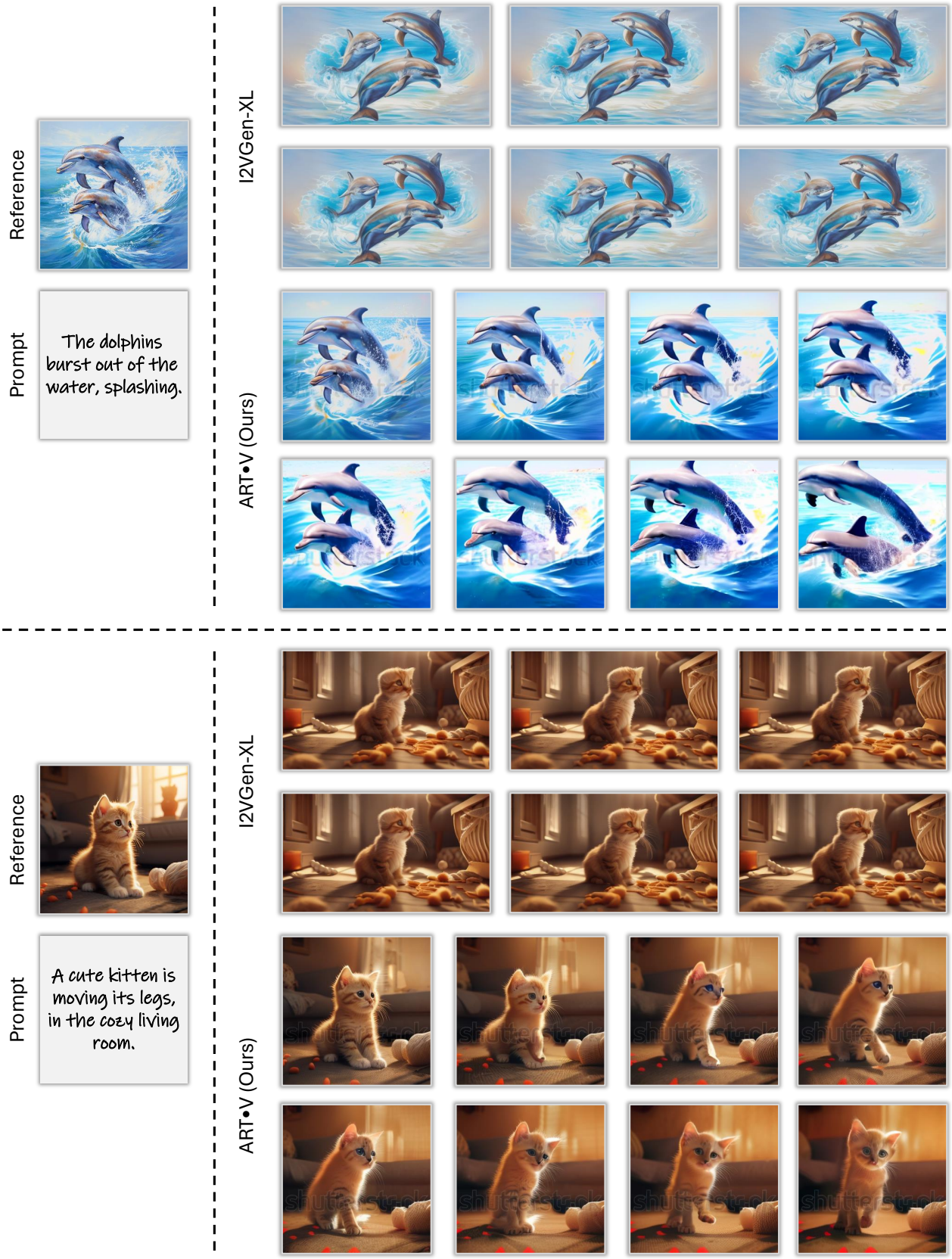

Figure 5. Visual results of text-image-to-video generation.

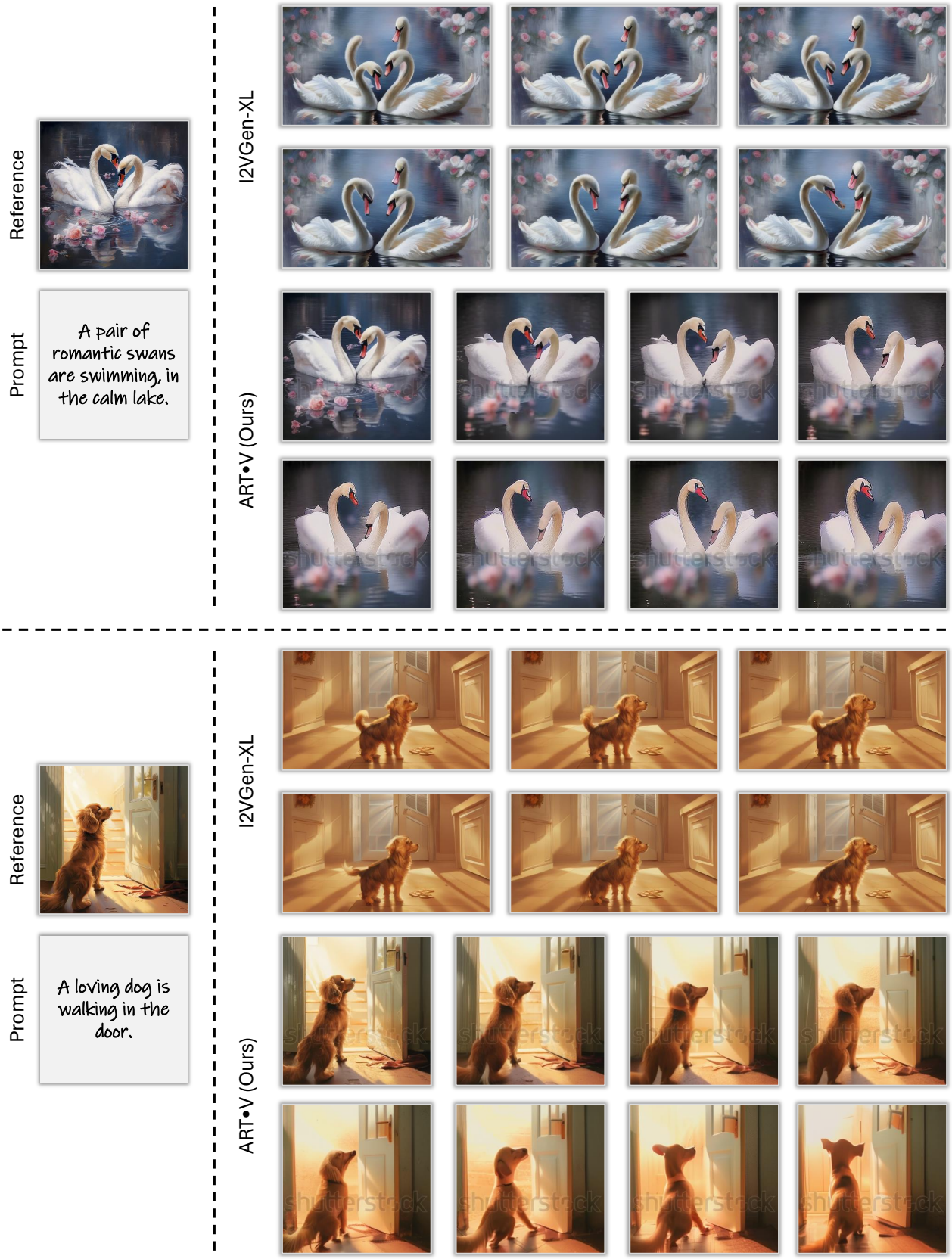

Figure 6. Visual results of text-image-to-video generation.

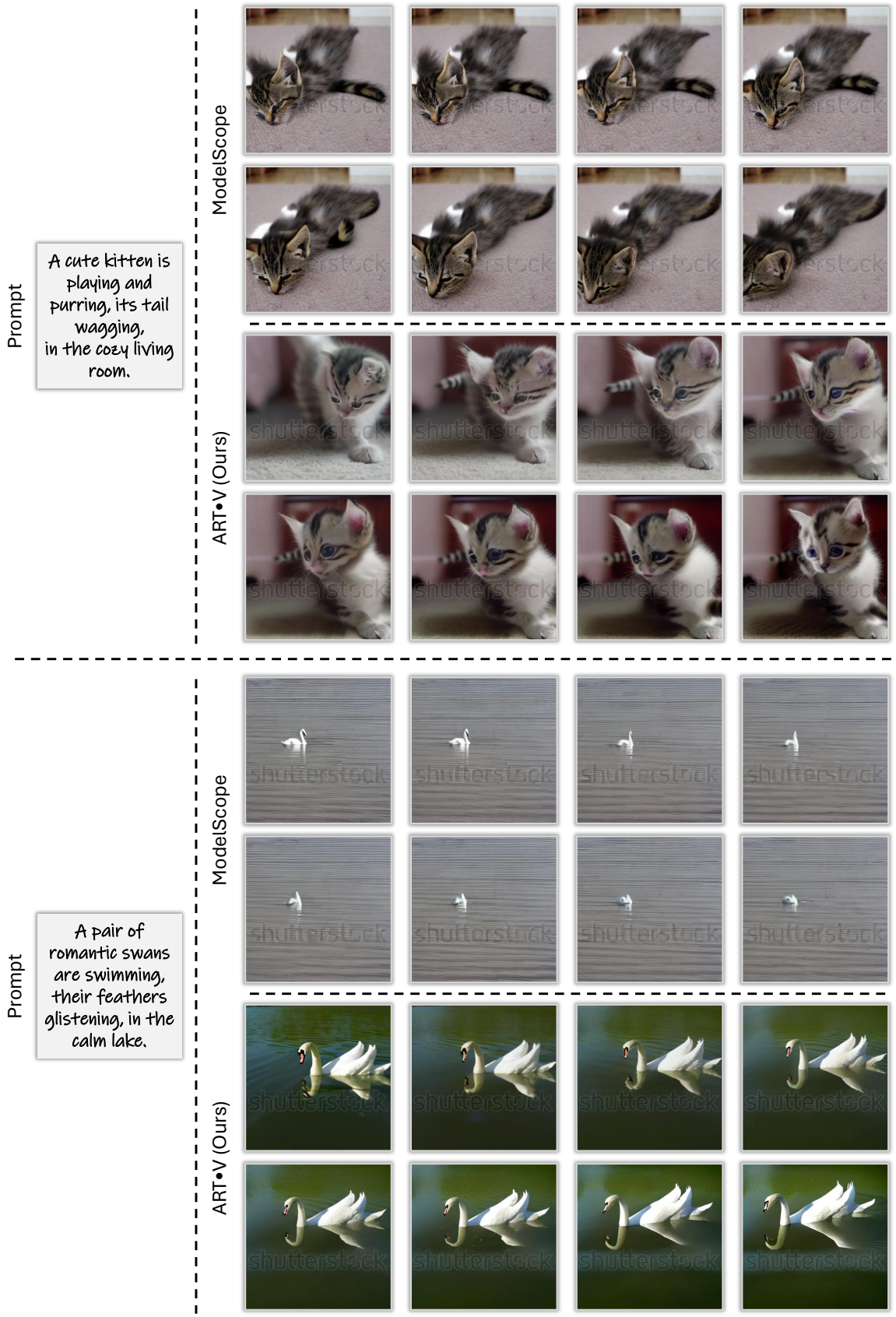

Figure 7. Visual results of text-to-video generation.

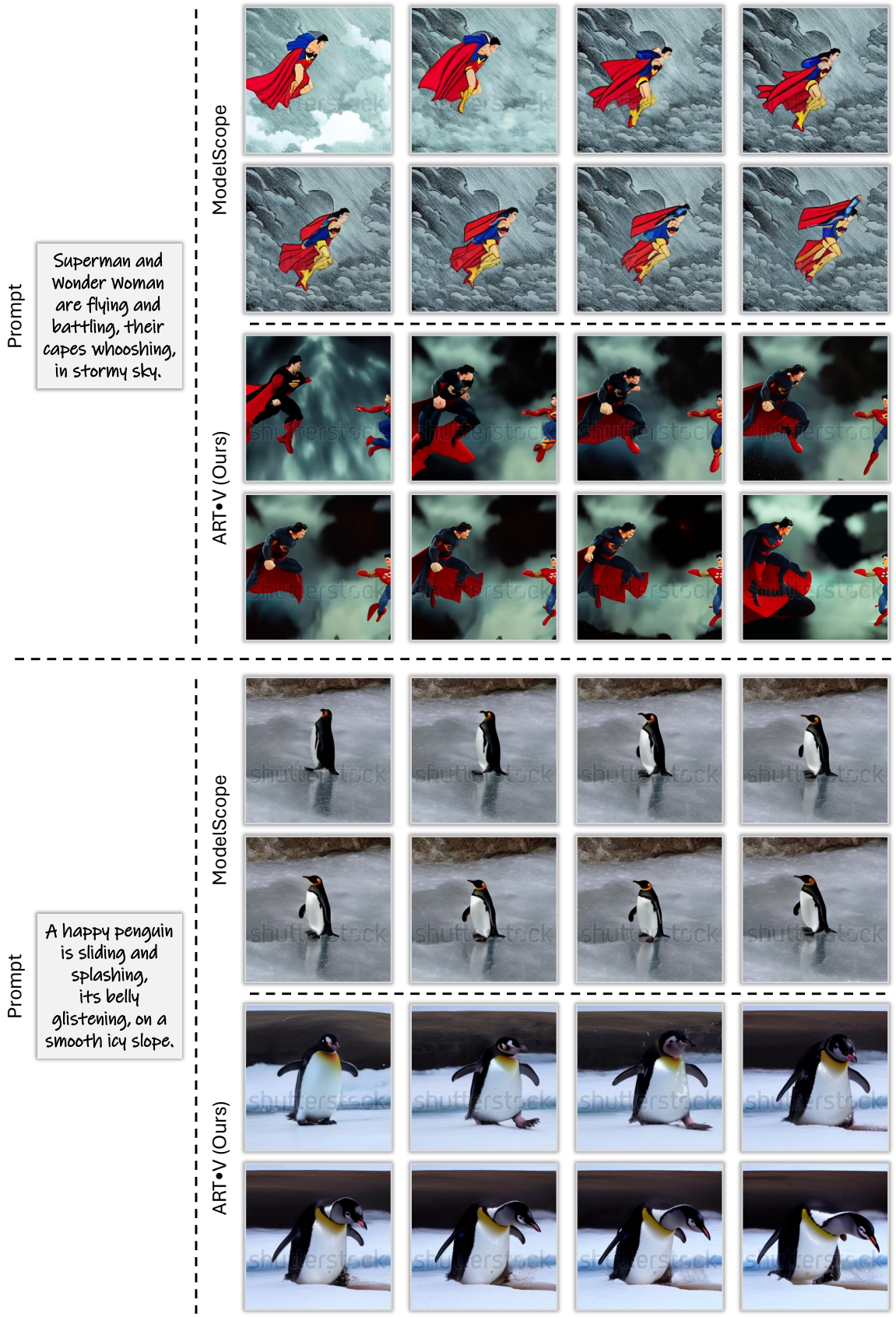

Figure 8. Visual results of text-to-video generation.

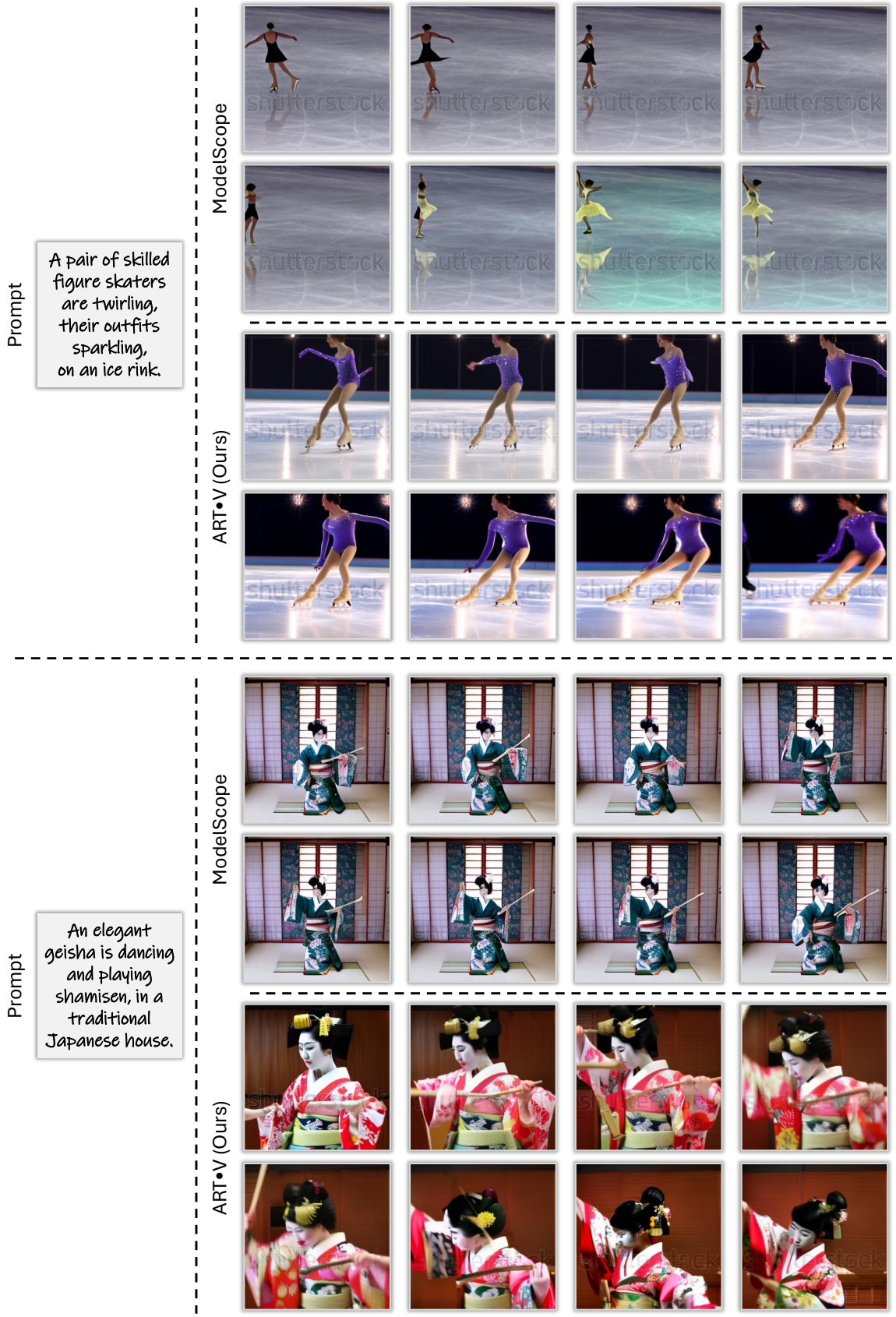

Figure 9. Visual results of text-to-video generation.

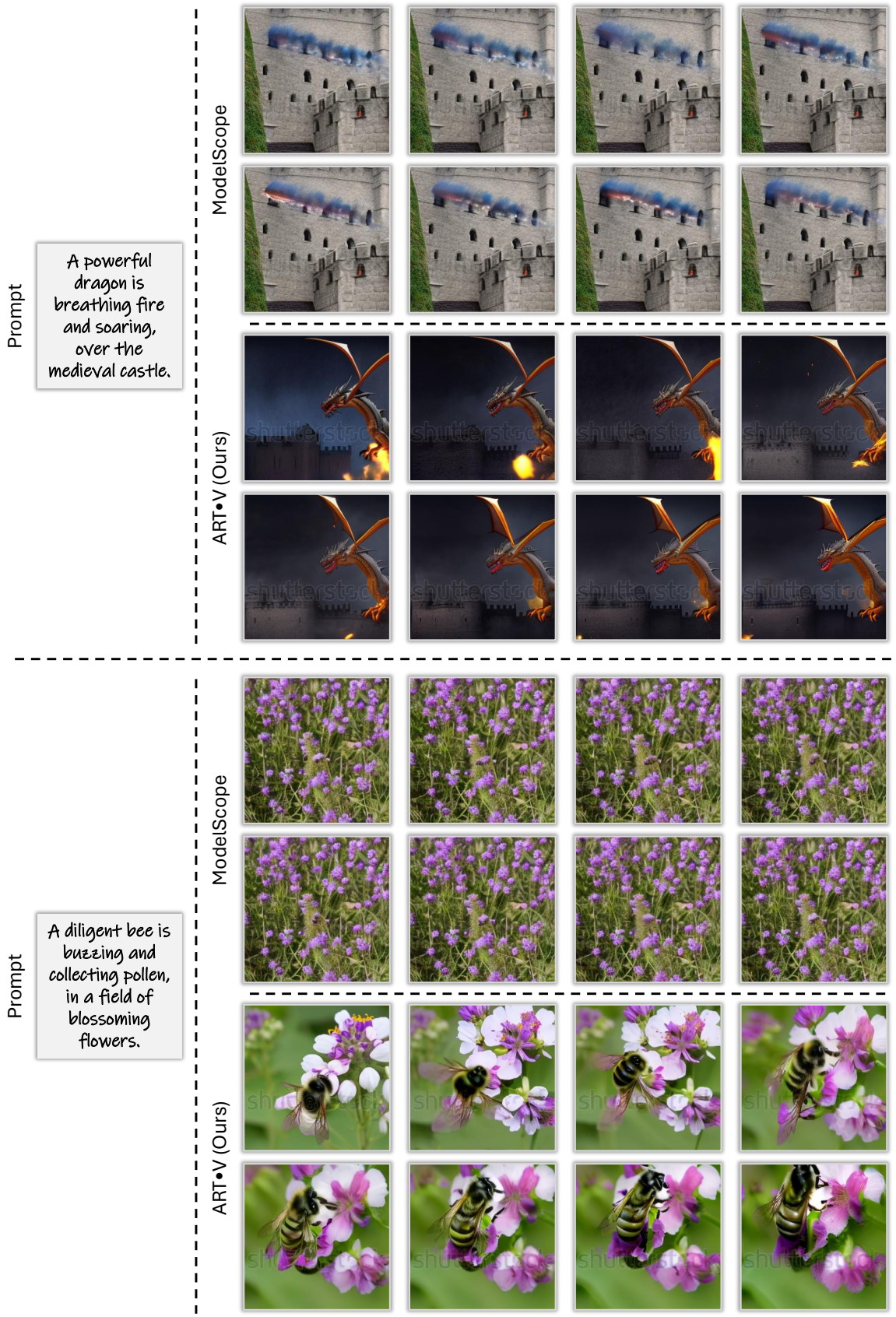

Figure 10. Visual results of text-to-video generation.

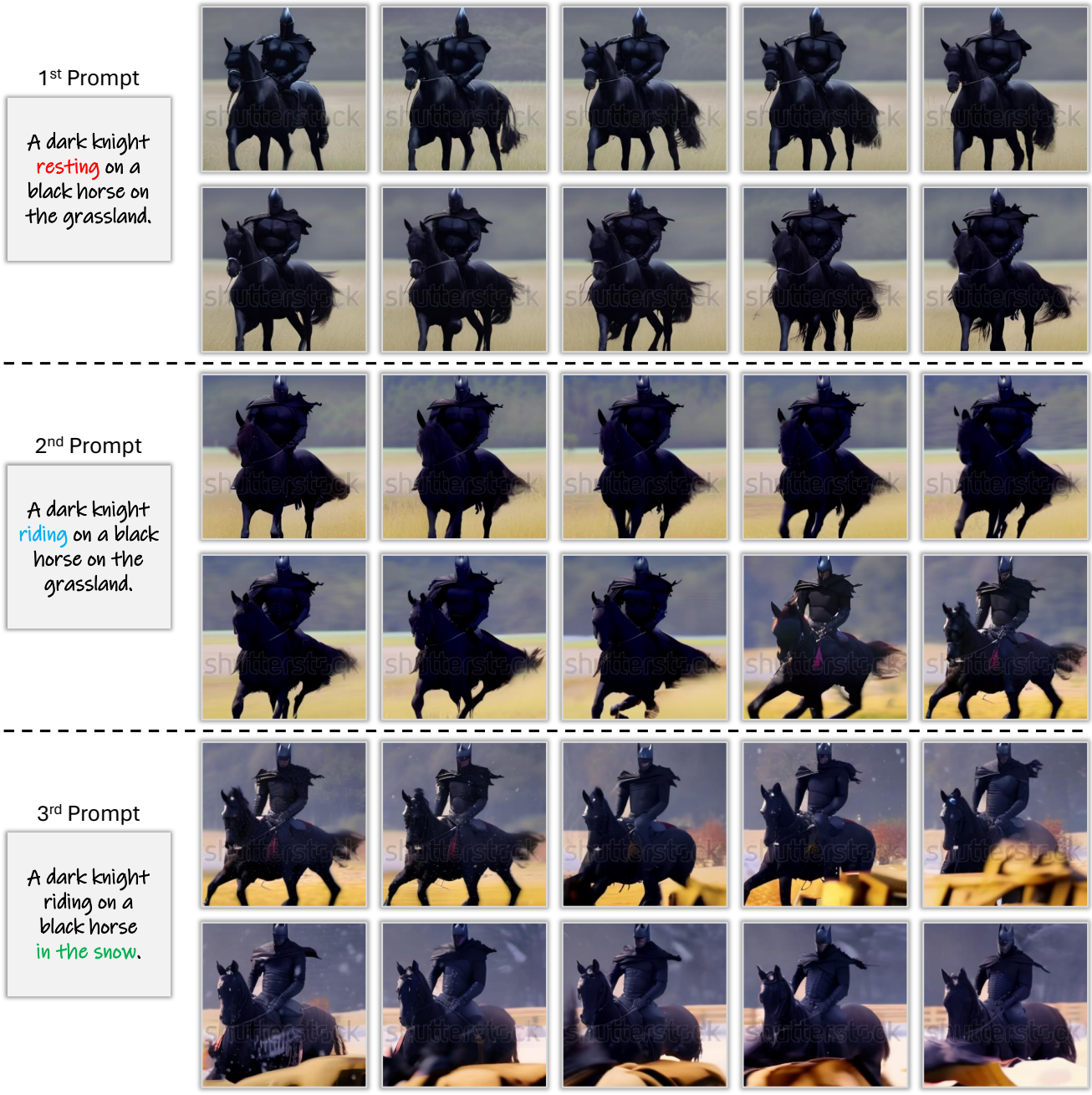

Figure 11. Visual results of multi-prompt text-to-video generation.

References

[1] https://modelscope.cn/models/damo/Image-to-Video/summary. 2

[2] https://www.midjourney.com/home. 2

[3] https://github.com/TencentARC/T2I-Adapter. 1

[4] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[5] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 1

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[9] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2