VideoBooth: Diffusion-based Video Generation with Image Prompts Supplementary Materials

Yuming Jiang[1] Tianxing Wu[1] Shuai Yang[3] Chenyang Si[1]
Dahua Lin[2] Yu Qiao[2] Chen Change Loy[1] Ziwei Liu[1✉]
[1]S-Lab, Nanyang Technological University [2]Shanghai AI Laboratory
[3]Wangxuan Institute of Computer Technology, Peking University

In this supplementary file, we will explain some implementation details in Section A. More details on comparison methods are introduced in Section B. In Section C, we will have more discussions on ablation study. In Section D, we will show more results, including our results and qualitative comparisons. Finally, in Section E, we will explain the watermark removal module.

A. Implementation Details

Data Processing. During the training, we will randomly sample 16 frames from original videos with a sampling rate of 4. We will randomly apply a horizontal flip to videos. The videos are center-cropped. In our proposed VideoBooth dataset, the image prompt is obtained by segmenting the subject from the first frame of a video. It should be noted that it is the first frame of the whole original video rather than the first frame of the video clip used for training. This avoids the image prompt being exact as the first frame during the training. However, even with this operation, the pose and scales of the subject in the first frame of the training clip are still similar to those of the image prompt. Also, visible parts of image prompts are almost the same as those in the first frame of training clips. This would lead to the model learning trivial solutions that the generated video has the exact viewpoint, pose, and scale as the image prompt. To alleviate this problem, we do some data augmentation to image prompts during the training. 1) To make the visible parts of image prompts to be different from those of training clips, we randomly crop image prompts. We cut off the edges of image prompts. The probability of cutting off a certain edge is set as 25%. For each edge, we randomly crop a portion of the border. The ratio of cropped parts is uniformly sampled from 0.01 to 0.2. 2) To introduce diverse scales and positions in image prompts, we perform random affine transformations to image prompts. We use the "transforms.RandomAffine" function in PyTorch. The parameter "degrees" is set as 30. The image will randomly rotate in the range of 0 to 30 degrees. The parameter "translate" is set as $(0.1, 0.1)$. The image will be horizontally shifted in the range of $(-0.1 * \text{img\_width}, 0.1 * \text{img\_width})$, and vertically shifted in the range of $(-0.1 * \text{img\_height}, 0.1 * \text{img\_height})$. The parameter "scale" is set as $(0.8, 1.2)$, which specifies the scale factor. Additionally, the parameter "fill" is set as 255, resulting in the exterior of the transformed image being filled with a white color. 3) We horizontally flip the image. The probability of flipping the image is set as 0.5. In future work, we plan to use image-to-3D models [3, 6] to augment views of image prompts.

Training. Our proposed VideoBooth injects the image prompt in a coarse-to-fine manner. The full model is trained in two stages. In the coarse stage, the model is trained using 8 GPUs. The batch size per GPU is set as 2. The global batch size is 16. We use the AdamW [4] optimizer. The learning rate is set as 1×10^{-4}, and the weight decay is set as 0. The model is initialized with the pretrained global mapper of ELITE [7]. For the fine stage, we use the model from the coarse stage as the initialization. For the newly added weights, we inherit weights from the base video model. The K and V projections for the image prompt are initialized with the K and V projection of cross-frame attention. The fine stage is trained using 8 GPUs. The batch size per GPU is set as 2. The global batch size is 16. We use the AdamW [4] optimizer. The learning rate is set as 1×10^{-4}, and the weight decay is set as 0. At the fine stage, for the retention of the capability of classifier-free guidance, we randomly set image prompts as null images, *i.e.* images filled in black, and text prompts are set as null text. The probability of setting these conditions to null is 0.1.

B. Comparison methods

Textual Inversion. In Textual Inversion [2], the appearance of target subjects is embedded into the text embeddings. A text token S^* is optimized to represent one specific subject. When applied to text-to-image models, multiple images containing the same object are required to optimize the text token S^*. In the setting of video generation, we directly use multiple video clips split from the original long video to optimize the text token. Once optimized, the text token S^* is used to replace the word embeddings of the target subject in the sentence to sample new videos.

DreamBooth. In DreamBooth [5], target subjects are injected into text tokens and model weights simultaneously. During the training, both model weights and a special token $S*$ are optimized. Similar to Textual Inversion, we use the original video and text description to train the model. Multiple video clips sampled from the long video are employed to optimize weights and text token $S*$. Once trained, the text token S^* is inserted before the word embeddings of the target object to sample new videos.

ELITE. Different from Textual Inversion and DreamBooth, ELITE [7] is an encoder-based method for fast customized generation. An encoder is trained to transform the images into embeddings. Local mapping and global mapping are employed to transform the CLIP embedding of image prompts into the features, which are fed into the cross-attention module. We adapt ELITE to video generation. We train the model using same data and same base video model.

C. More Discussions on Ablation Study

In this section, we show one more visual example of the ablation study. In Fig. 7 of the main paper, we show that the model with only coarse embeddings results in imprecise encoding of appearance. Both the model trained with only fine embeddings and the model trained using the unified training strategy overfit to image prompts. In the two examples shown in Fig. 7 of the main paper, the first frames can take the image prompt, but the generated appearance is distorted along the frames. In Fig. A2, we discuss another case, which exhibits a different behaviour.

Only Coarse Embeddings. This ablation model injects image prompts with only coarse embeddings via Image Encoder. As shown in Fig. A2(a), coarse embeddings provide coarse but not precise guidance. The face of the dog and the shape of the head are not accurately captured. Our full model shown in Fig. A2(d) can capture visual details.

Only Fine Embeddings. In this ablation model, we only have fine embeddings of image prompts in cross-frame attention layers. Recall that the purpose of fine embeddings is to refine the encoding from coarse levels. In the example shown in Fig. A2(b), the first frame does not successfully embed the image prompt. Without coarse embeddings, the generation of the appearance of the dog relies purely on the propagation from the first frame. The failure in encoding the image prompt into the first frame results in the following frames having random appearances for the dog.

The Necessity of Coarse-to-Fine Training. In VideoBooth, we propose the coarse-to-fine training strategy, *i.e.*, train the coarse embeddings first and then train the attention injection module. This ablation model is trained within one stage. In the example shown in Fig. A2(c), the first frame successfully takes the appearance of the image prompt. The model generates a consistent appearance in all frames, but the motion of this generated clip is small and not aligned with the text prompt. We found that in the case of generating small motions or static frames, the coarse-to-fine training strategy can work well. However, when it comes to generating large motions as shown in Fig. 7 of the main paper, the appearance will be distorted along frames.

D. More Qualitative Results

In Fig. A3 and Fig. A4, we show several groups of results generated by our proposed VideoBooth. In each group, video clips are generated using the same image prompt but different text prompts. In Fig. A5, we show two groups of results. In each group, video clips are generated using the same text prompt but different subjects as specified in the image prompts. In the first group, our model is capable of differentiating the characteristics of pandas and reflecting them in the generated videos. In the second example, our model can generate several video clips for different cats looking at the laptop. In Fig. A6, A7, A8, we show more qualitative comparisons with baseline methods.

E. WaterMark Removal Module

Since the videos in WebVid dataset [1] have a watermark, the model trained using this dataset generates videos with a watermark in nature. To generate videos without watermark for better visual quality, we finetune the model with an additional module using the Vimeo dataset [8]. We only use text prompts and original videos to finetune the model. As shown in Fig. A1, we add six blocks before the last conv out layer of the base video model. The added six blocks can be regarded as a small UNet. After the first block, we downsample features by two times. Then after the second block, features are downsampled by two times. Then features are enhanced by two blocks. Finally, features are upscaled with two consecutive blocks. Each block upsamples features by two times. Inside each block, there are two ResNet blocks. Skip connections are adopted between downsampling blocks and upsampling blocks. After all blocks, we feed the model to one conv layer, which is initialized with zero. The motivation for zero initialization is to avoid the newly added blocks affecting the model. We add the obtained features to the original features as residues. The added features are fed into the final layer (*i.e.* Conv Out layer) of the base video generation model. The newly added modules and the last layer are optimized during the finetuning. After finetuning, the watermark can be removed without influencing the generative capability of VideoBooth. It should be noted that we use the model without watermark removal module when comparing with baselines.

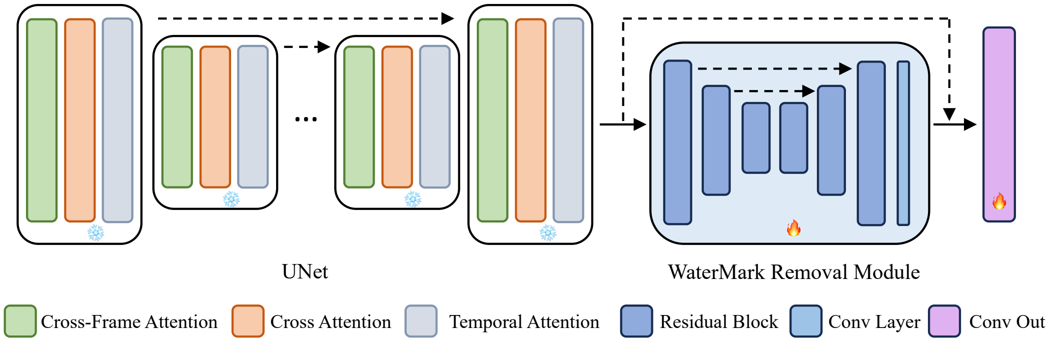
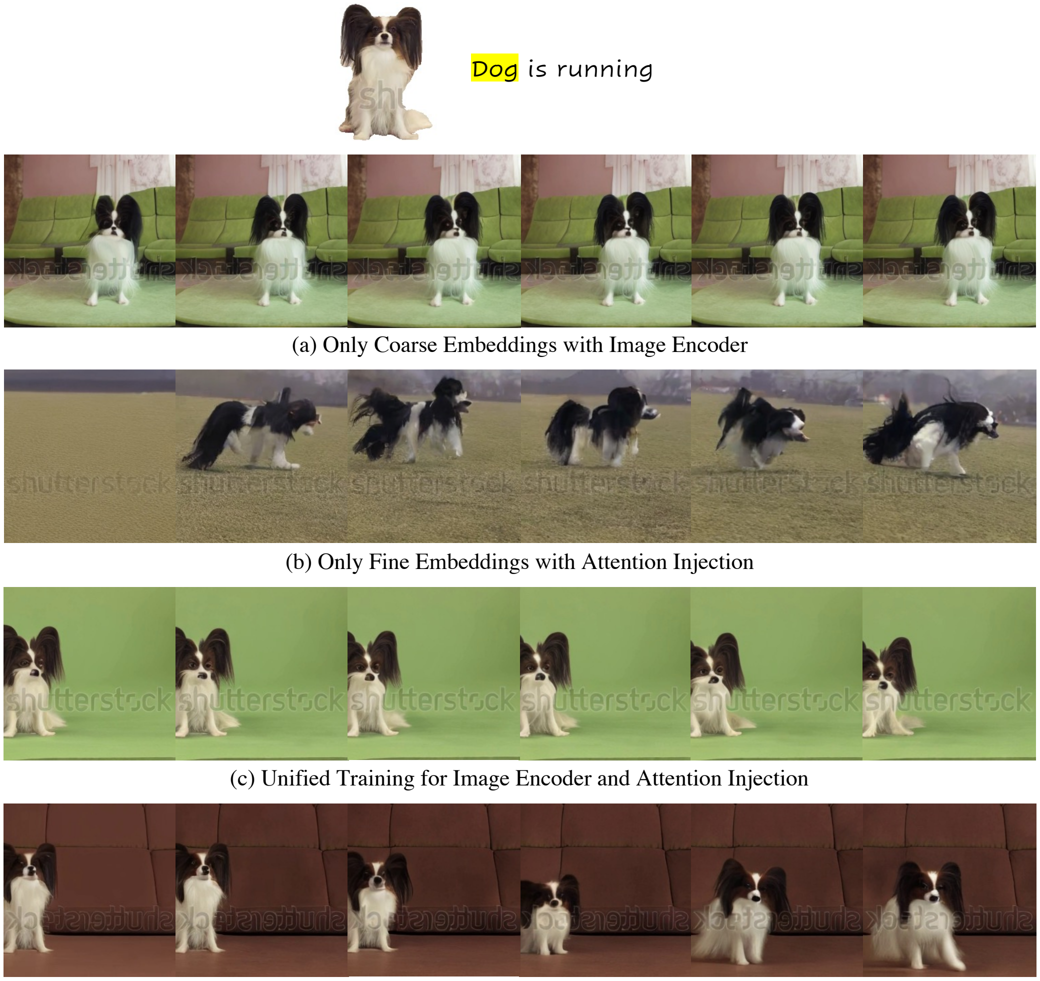


Figure A1. **Illustration of Watermark Removal Module.** We add a WaterMark Removal Module before the conv out layer. The output of the watermark removal module is added as a residue to the original features. We finetune the newly added module and the conv out layer using the video data [8] without watermarks.



(a) Only Coarse Embeddings with Image Encoder

(b) Only Fine Embeddings with Attention Injection

(c) Unified Training for Image Encoder and Attention Injection

(d) Full Model

Figure A2. **More Visual Analysis on Ablation Study.**

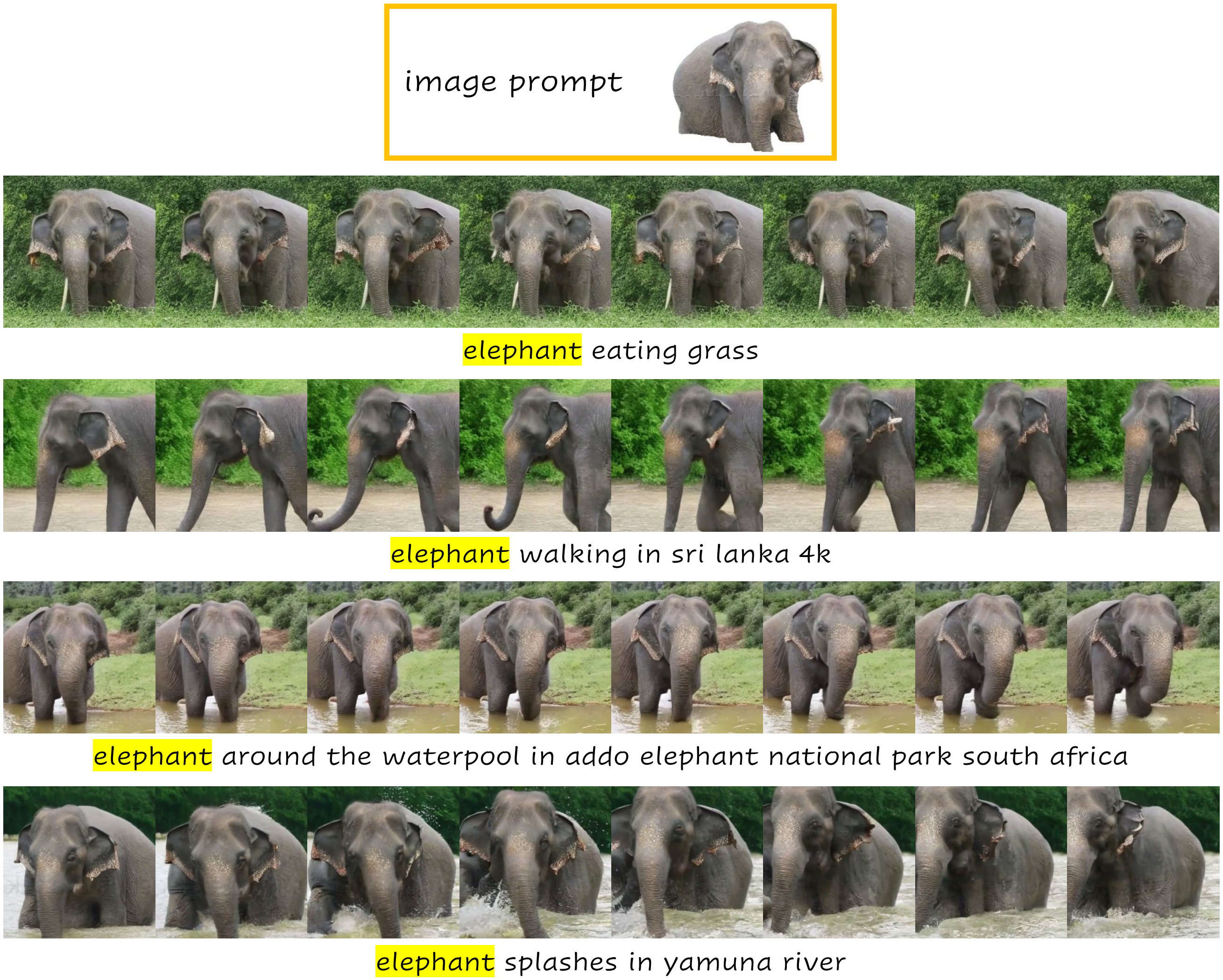


image prompt

elephant eating grass

elephant walking in sri lanka 4k

elephant around the waterpool in addo elephant national park south africa

elephant splashes in yamuna river

Figure A3. **More Visual Results of VideoBooth.**

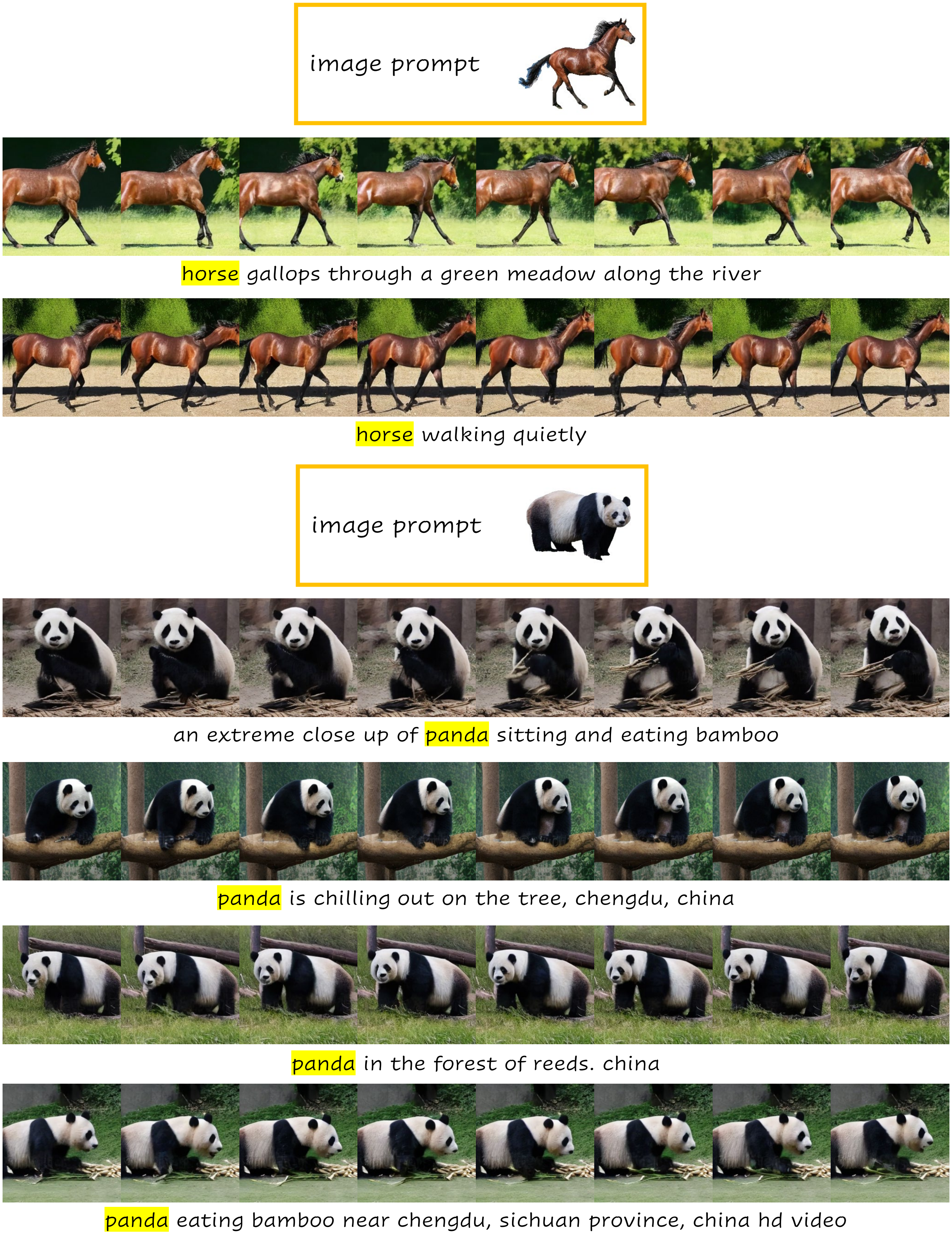


image prompt
horse gallops through a green meadow along the river
horse walking quietly
image prompt
an extreme close up of panda sitting and eating bamboo
panda is chilling out on the tree, chengdu, china
panda in the forest of reeds. china
panda eating bamboo near chengdu, sichuan province, china hd video

Figure A4. **More Visual Results of VideoBooth.**

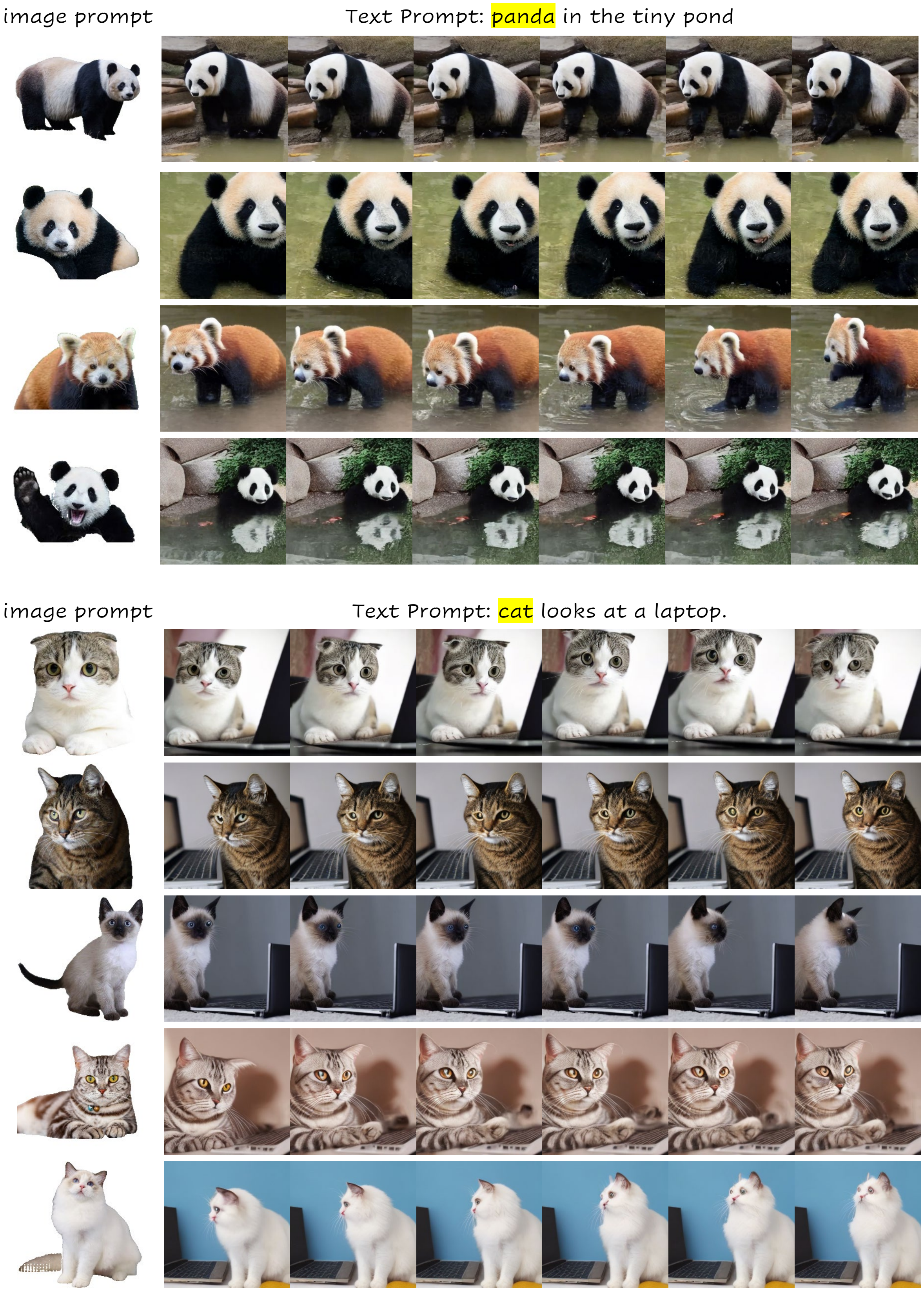


image prompt
Text Prompt: panda in the tiny pond
image prompt
Text Prompt: cat looks at a laptop.

Figure A5. **More Visual Results of VideoBooth.**

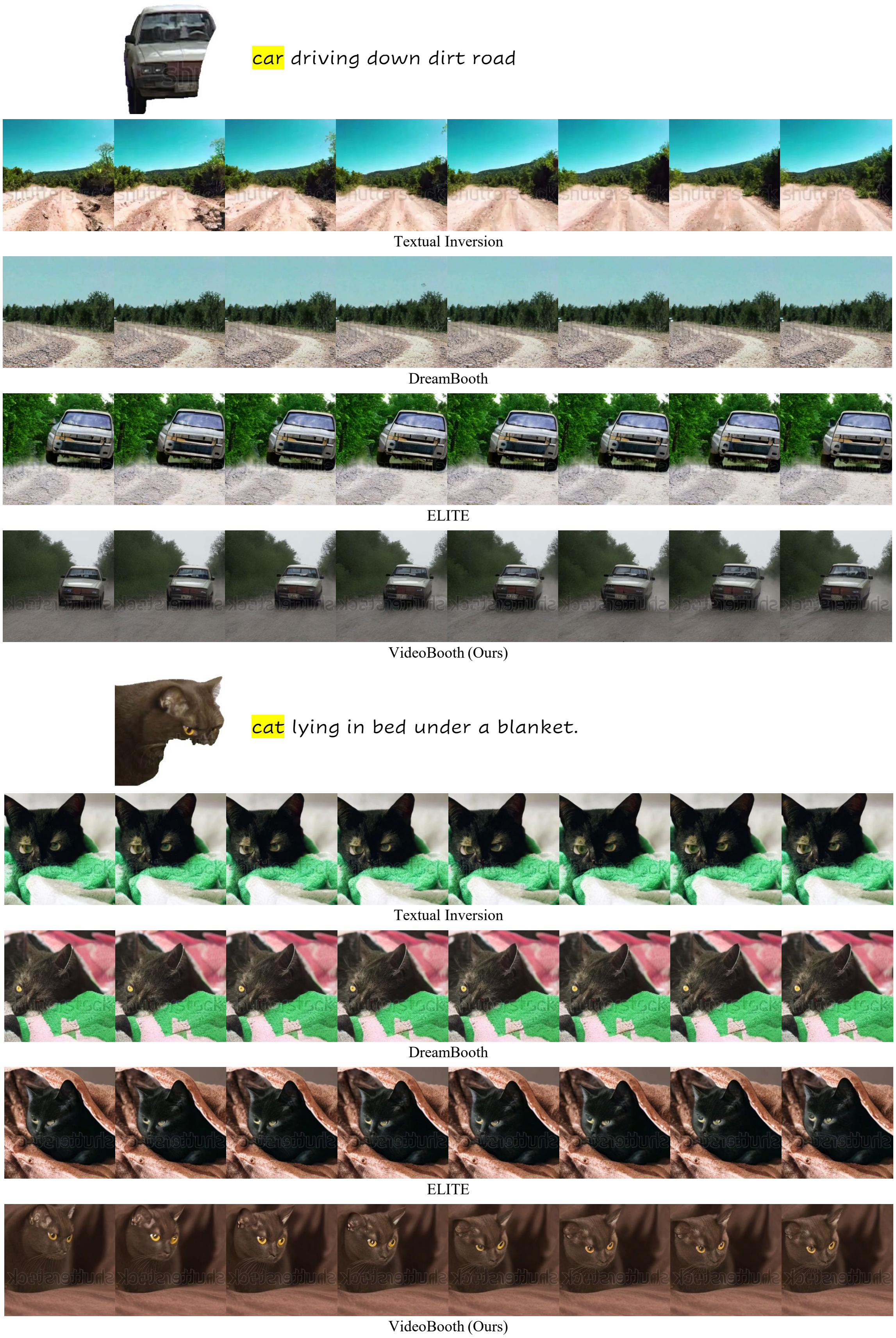
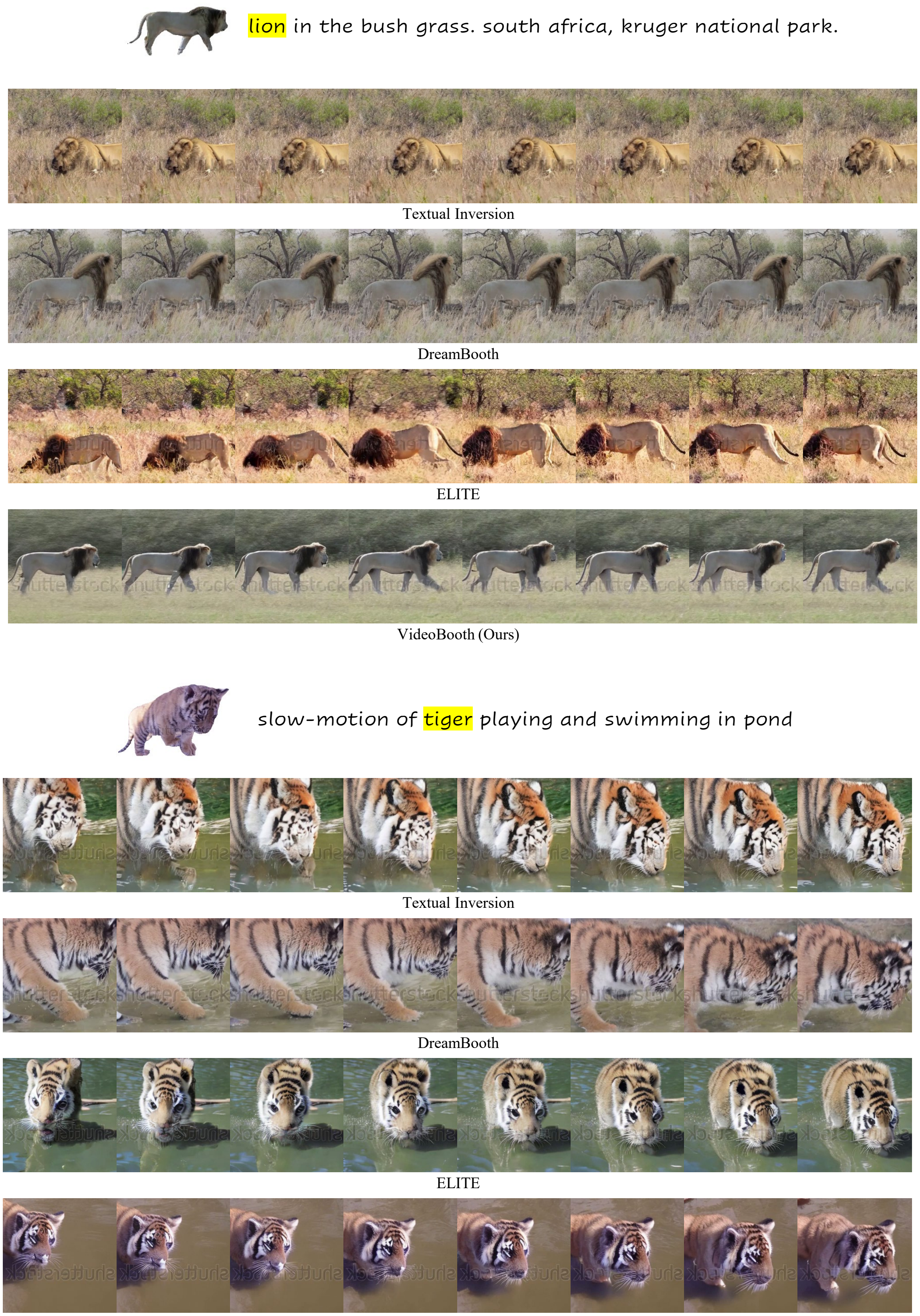


Figure A6. **Qualitative Comparison with Baseline Methods.**


lion in the bush grass. south africa, kruger national park.
slow-motion of tiger playing and swimming in pond

Textual Inversion

DreamBooth

ELITE

VideoBooth (Ours)

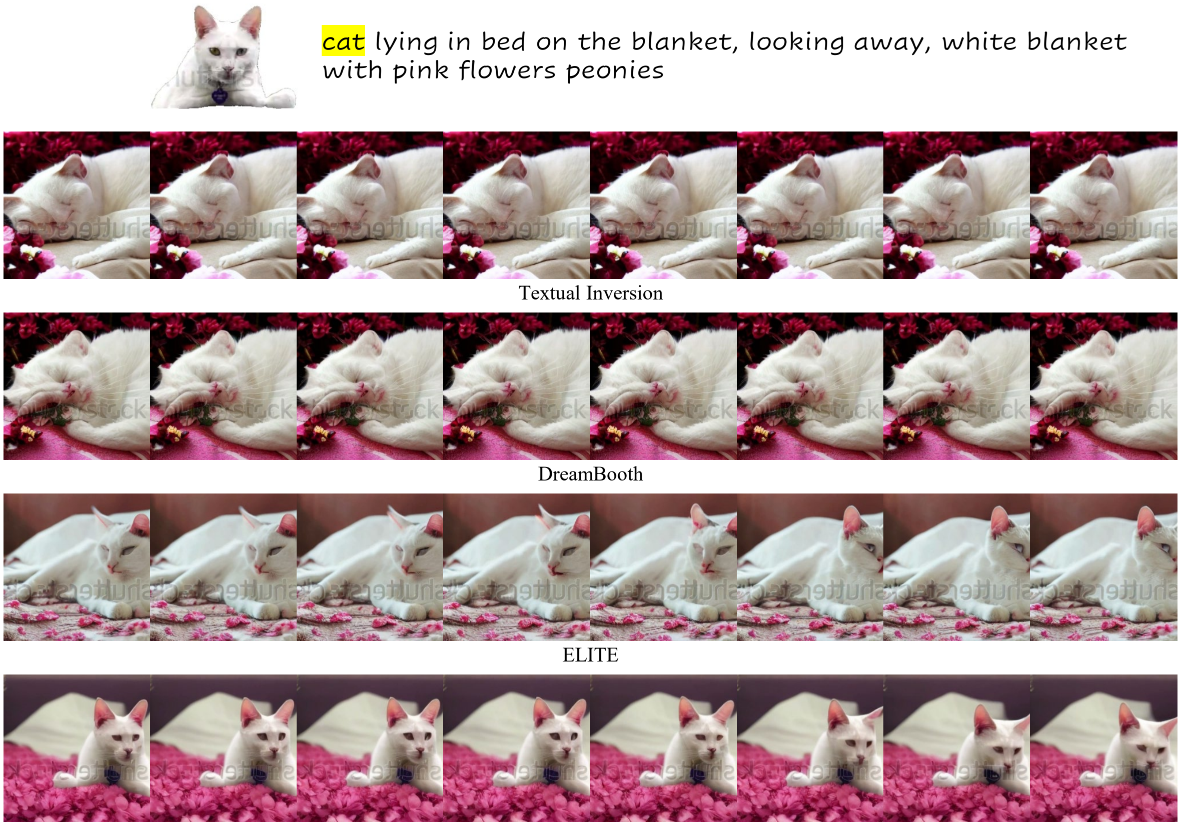
Textual Inversion

DreamBooth

ELITE

VideoBooth (Ours)

Figure A7. **Qualitative Comparison with Baseline Methods.**


cat lying in bed on the blanket, looking away, white blanket with pink flowers peonies

Textual Inversion

DreamBooth

ELITE

VideoBooth (Ours)

Figure A8. **Qualitative Comparison with Baseline Methods.**

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 2

[2] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 1

[3] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 1

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[5] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022. 2

[6] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1

[7] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. *arXiv preprint arXiv:2302.13848*, 2023. 1, 2

[8] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019. 2, 3