Multi-subject Open-set Personalization in Video Generation

Supplementary Material

A. Details of Training Datasets and Augmentations

A.1. Training Datasets and Undesirable Samples Filtering

Our personalization training dataset is built on Panda-70M [13] and other internal video-caption datasets, consisting of 86.8M videos. However, the original dataset includes undesirable video samples for video generation. We classify these undesirable samples into four categories:

- Still foreground image: a video with only pan and zoom effects of a static image.
- Slight motion: a video with tiny camera movement and static foreground objects
- Screen-in-screen: a video with an image or video overlaying on a background image or video.
- Computer screen recording: a video depicting a screen recording (excluding PC games).

To filter out these samples, we learn a video classification model. Specifically, we randomly sample 40k videos from our training dataset and manually annotate them based on the above criteria. Using these labels, we fine-tune VideoMAE [69] for video classification. Moreover, as we aim to generate single-shot videos, we apply TransNetV2 [65] to detect and exclude videos that contain multiple shots. We only retain the desirable single-shot videos for training.

A.2. Retrieving Entity Words

In Section 3.1, we use a large language model [31] (LLM) to retrieve the entity words from the caption, using the instruction template shown in Figure 7.

Given an image caption, please retrieve the entity words that indicate background, subject, and visually separable objects.
[Definition of background] the background spaces that appear in most of the image area.
[Definition of subject] human or animal subjects that appear in the image.
[Definition of object] the entities that appear in part of the image and can be visually separated with each other.
All entity words need to strictly follow two rules below:
1) the entity word is a noun without any quantifier.
2) the entity word is an exact subset of the caption. Do not modify any characters, words, and symbols.
Here are some examples, follow this format to output the results:
Caption: A woman in a mask and coat, with long brown hair, shows a small green-capped bottle to the camera.
Output: {'background': [''], 'subject': ['woman'], 'object': ['mask', 'coat', 'long brown hair', 'green-capped bottle']}

(More examples)

Figure 7. **Prompt template for retrieving the entity words.**

Given the caption, the LLM extracts a list of entity words, with the following steps.

- Remove the sample if the output of the LLM is not in a valid dictionary format.
- Remove the sample if any entity word is not a sub-string of the caption.
- Reclassify the entity words according to the pre-defined rules. For example, “cloud” is not a visually separable object and is supposed to be classified into a background entity word.
- Remove the sample with no subject entity word, as we observe that the video motion of these samples is typically trivial camera movements and lacks meaningful foreground motion.
- Remove the sample with the subject entity word in the plural form, as this will introduce ambiguity when applying the localization algorithm.

We curate a training dataset comprising 37.8M videos. To illustrate the diversity of subjects within our dataset, we plot a word cloud of entity words from 10k randomly sampled training videos in Figure 8.

Figure 8. **Word cloud of the entity words.** We randomly sample 10k videos from our training dataset and plot the word cloud of the retrieved subject and object entity words.

Table 4. **Training augmentations.** We denote the height and width of the reference image as h and w.

	Apply Probability	Hyperparameters	
		Type	Sampling Range
Downscale	1.0	scale	$[112/\max(h,w), 1.0]$
Gaussian blur	1.0	kernel size (px)	$[1, \max(h,w)/50]$
Color jitter	1.0	scale	$[-0.05, 0.05]$
Brightness	1.0	scale	$[0.9, 1.1]$
Horizontal flip	0.5	-	-
Shearing (x-axis)	1.0	value (px)	$[-0.05, 0.05] \times w$
Shearing (y-axis)	1.0	value (px)	$[-0.05, 0.05] \times h$
Rotation	1.0	value (°)	$[-20, 20]$
Random crop	1.0	scale	$[0.67, 1.0]$

A.3. Data Augmentation and Conditional Images Sampling

In Section 3.3, we introduce data augmentation to prevent models from overly relying on the undesirable properties of the reference image. Table 4 lists the applied augmentation. While augmentations can reduce model overfitting to some extent, we observe that models could also overfit to the number of reference images. Specifically, if we always use all available reference images as conditions during training, the model can generate the target subject with some properties correlated to the number of reference images (*ref*) during inference. Using the text prompt "*A dog is running*" as an example:

- If users input 0 *ref*, the model generates a tiny or heavily occluded dog.
- If users input 1 *ref*, the model generates a dog that is running out of view of the video.
- If users input 3 *ref*s of a similar pose, the model generates a dog that is running in slow motion.

To avoid models overfitting on the number of the reference images, we design a sampling algorithm to select conditional subjects and their reference images during training. It includes the following five steps:

- Randomly sample the number of conditional subjects from 1 to 3.
- Randomly sample conditional subjects with replacement.
- For each subject, randomly sample the number of conditional reference images from 1 to 3.
- For each subject, randomly sample conditional reference images with replacement.
- Randomly include background conditioning with a probability of 50%.

Table 5. **Architecture details of video generation backbone and image encoders.**

Video Backbone	DiT [53]
Input channels	16
Patch size	$1 \times 2 \times 2$
Latent token channels	4096
Positional embeddings	RoPE
DiT blocks count	32
Attention heads count	32
Use flash attention	✓
Use fused layer norm	✓
Use self conditioning	✓
Conditioning channels	1024
Conditioning images	6 (stage II training only)

Image Encoder	CLIP [55]	DINOv2 [50]
Architecture	ViT-L/14	ViT-L/14
Selective block	23	24
Selective tokens	patch	patch
Tokens count	256	256
Tokens channels	1024	1024

Table 6. **Training hyperparameters.** The right table is for stage II training.

Stage	I	II
Steps	60k	40k
Warmup steps	-	1k
Samples seen	234M	39M
Image conditioning	✗	✓
Optimizer	AdamW	
Learning rate	$1e^{-4}$	
LR scheduler	constant	
Beta	$[0.9, 0.99]$	
Weight decay	0.01	
Gradient clipping	0.05	
Dropout	0.1	

# frames	Batch Size (Sampling Weight)		
	256px × 144px	512px × 288px	1024px × 576px
17	1,216 (10%)	304 (10%)	80 (10%)
49	464 (3.3%)	112 (5.8%)	32 (5.8%)
73	320 (3.3%)	80 (5.8%)	16 (5.8%)
97	240 (3.3%)	64 (5.8%)	16 (5.8%)
121	192 (3.3%)	48 (5.8%)	16 (5.8%)
145	160 (3.3%)	- (0%)	- (0%)
193	128 (3.3%)	- (0%)	- (0%)
289	80 (3.3%)	- (0%)	- (0%)

B. Details of Model Architecture, Training, and Inference

B.1. Model Architecture

Our framework is a latent-based diffusion model. We use CogVideoX-5B [84] as the autoencoder with a compression rate of $4 \times 8 \times 8$ in temporal, height, and width dimensions. We use DiT [53] as the video backbone with two different image encoders, including CLIP [55] and DINOv2 [50]. We detail the hyperparameters of the video backbone and image encoders in Table 5. For the video backbone, we follow the original DiT designs to embed input timesteps using adaLN-Zero block, which is composed of adaptive normalization layers [54] with scaling parameters α that are applied immediately prior to any residual connections within the DiT block. For the image representations, we find that using the patch tokens as the image embeddings can retain more localized properties of the reference images and result in higher fidelity than the class token.

B.2. Model Training

We present the training details of the model in Table 6. We train the model in two stages. In the first stage, we fix the autoencoder and train the video backbone without the cross-attention layer for personalization for 60k steps. In the second stage, we introduce the cross-attention layer for personalization and fine-tune the model for additional 40k steps. With more details, in the second stage, we apply a 1k-step linear warmup and only train the newly introduced cross-attention layer while keeping the video backbone fixed at the first 10k steps. For the following 30k steps, we fine-tune the entire video model with the image encoder frozen. We use the AdamW [43] optimizer with a constant learning rate of $1e^{-4}$. To achieve stable training, we set $\beta = [0.9, 0.99]$, a weight decay of 0.01, gradient clipping with the value of 0.05. We randomly drop text or image conditioning with a probability of 10% and set them to zero to support classifier-free guidance [27].

To enable the generation of high-resolution and long-duration videos while ensuring efficient model training, we train our model on videos of varying resolutions and lengths. Table 6 lists the batch size and sampling weights for the training videos across different resolutions and lengths. The batch size is set to balance the training time for each step with different attributes. We apply the fixed framerate of 24. Our model supports generating videos up to 12 seconds in length at $256\text{px} \times 144\text{px}$ resolution and up to 5 seconds in length at $512\text{px} \times 288\text{px}$ and $1024\text{px} \times 576\text{px}$ resolution.

Our model is implemented in PyTorch [52] and trained with 256 80GB A100 GPUs in stage I and 64 GPUs in stage II.

B.3. Model Inference

We use a rectified flow sampler [41] with classifier-free guidance [27] (CFG) for sampling. The choice of scale and implementation of the CFG can significantly impact the performance of diffusion models. Although our model performs best with a CFG scale of 8 for text conditioning, we find that applying such a large CFG scale for image conditioning can cause the model to replicate reference images directly into the video, without introducing natural motion and appearance variation. To address this, we follow Brooks *et al*. [5] and apply CFG twice within a sampling step, once for text conditioning and once for image conditioning, but with a slight change in CFG implementation. Formally, Brooks *et al*. [5] applies CFG as follows:

$$\tilde{e_\theta}(z_t, c_I, c_T) = e_\theta(z_t, c_I, c_T) + s_T \cdot (e_\theta(z_t, c_I, c_T) - e_\theta(z_t, c_I, \varnothing)) + s_I \cdot (e_\theta(z_t, c_I, \varnothing) - e_\theta(z_t, \varnothing, \varnothing)), \tag{1}$$

where $e_\theta(z_t, c_I, c_T)$ is the score estimation function with the image and text conditioning, denoted as c_I and c_T. We mark $c = \varnothing$ if we set condition c to zero. Empirically, we find that the formula below can achieve better visual quality in our case:

$$\tilde{e_\theta}(z_t, c_I, c_T) = e_\theta(z_t, c_I, c_T) + s_T \cdot (e_\theta(z_t, c_I, c_T) - e_\theta(z_t, c_I, \varnothing)) + s_I \cdot (e_\theta(z_t, c_I, c_T) - e_\theta(z_t, \varnothing, c_T)). \tag{2}$$

We set $s_T = 8$ and $s_I = 3$. We use 256, 128, and 64 denoising steps to synthesize the videos at $256\text{px} \times 144\text{px}$, $512\text{px} \times 288\text{px}$, and $1024\text{px} \times 576\text{px}$ resolution, respectively. Moreover, we apply time shifting [17, 21] to align the signal-to-noise ratio (SNR) at different resolutions.

C. More Visualization Results

In this section, we provide more synthetic samples to complement the evaluations. Appendix C.1 shows the samples of multi-subject and open-set customization. Appendix C.2 includes an ablation study in which we use different reference images to personalize the same conditional entity word from the same prompt. Appendix C.3 provides more comparisons with state-of-the-art personalization models on various conditional subjects.

C.1. Additional Results of Multi-subject Open-set Personalization

We show the multi-subject and open-set personalization samples in Figures 9 to 12. In each sample, we show the generated videos with one to three conditional subjects or backgrounds by incrementally increasing the number of reference images. In addition, we provide synthetic videos without reference images at the bottom to showcase the effect of image conditioning.

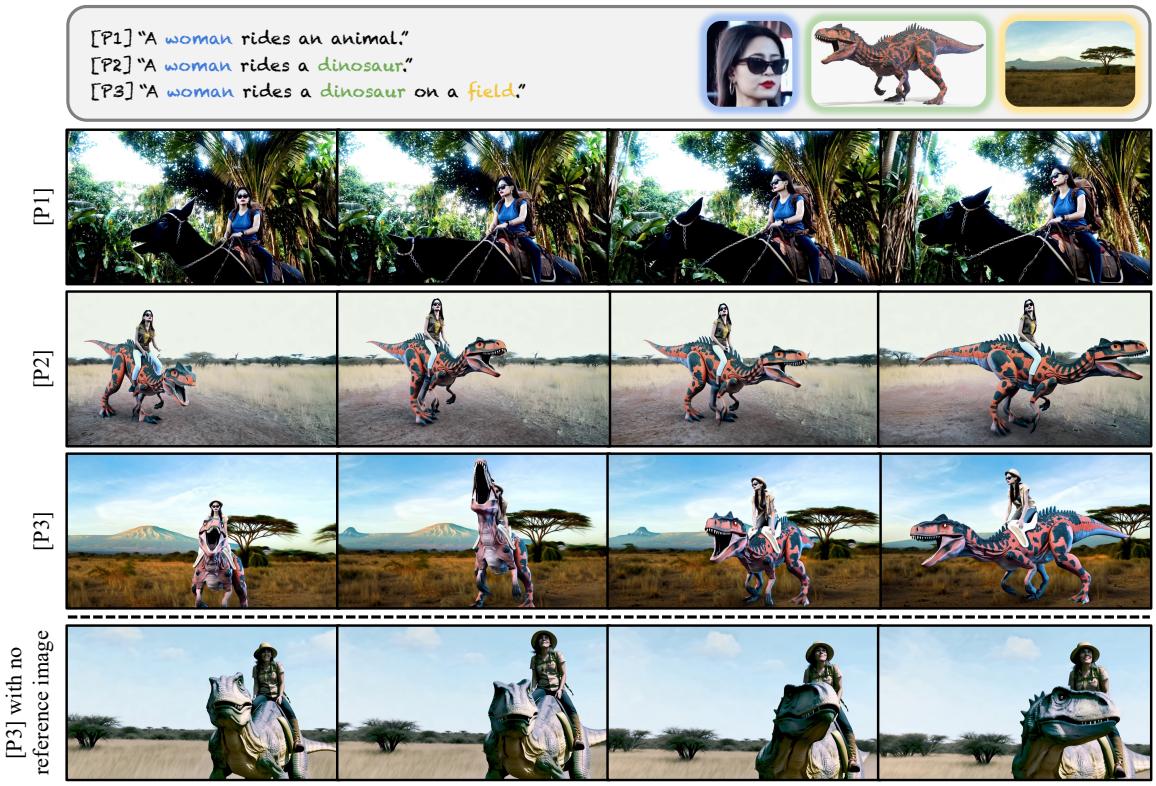

Figure 9. **Additional results of multi-subject open-set personalization.**

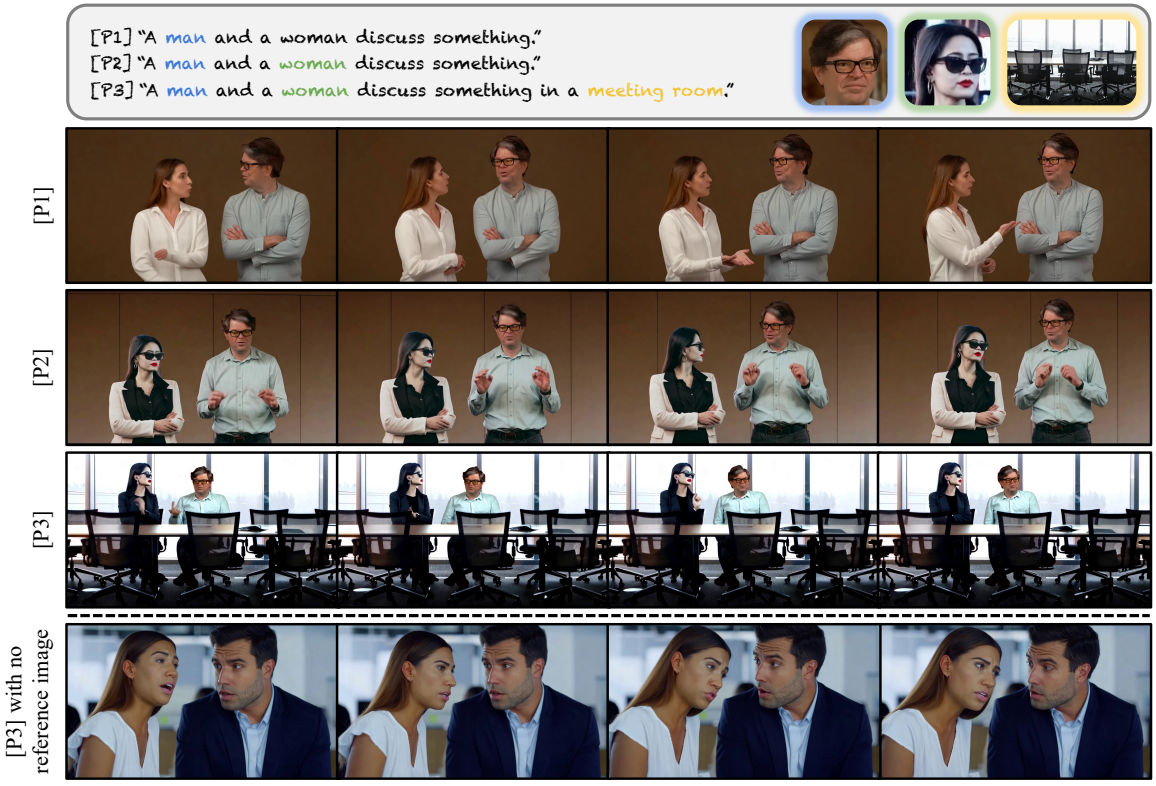

Figure 10. **Additional results of multi-subject open-set personalization.**

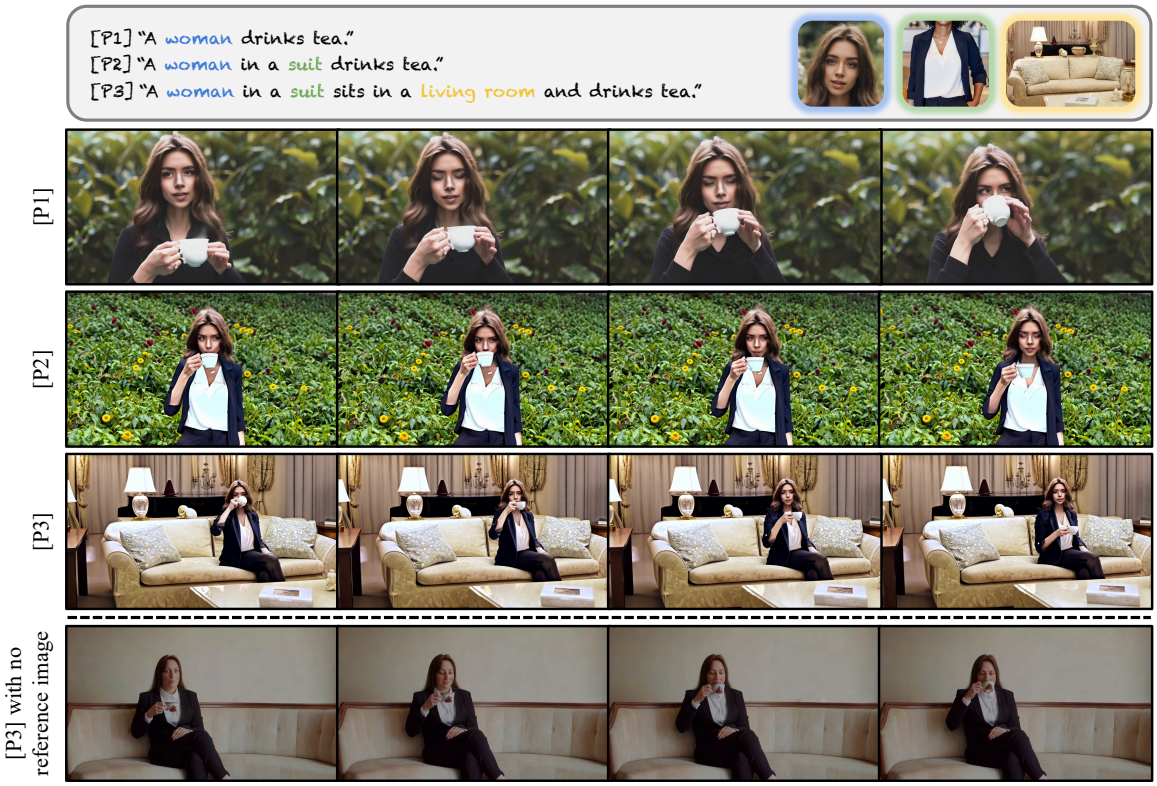

Figure 11. **Additional results of multi-subject open-set personalization.**

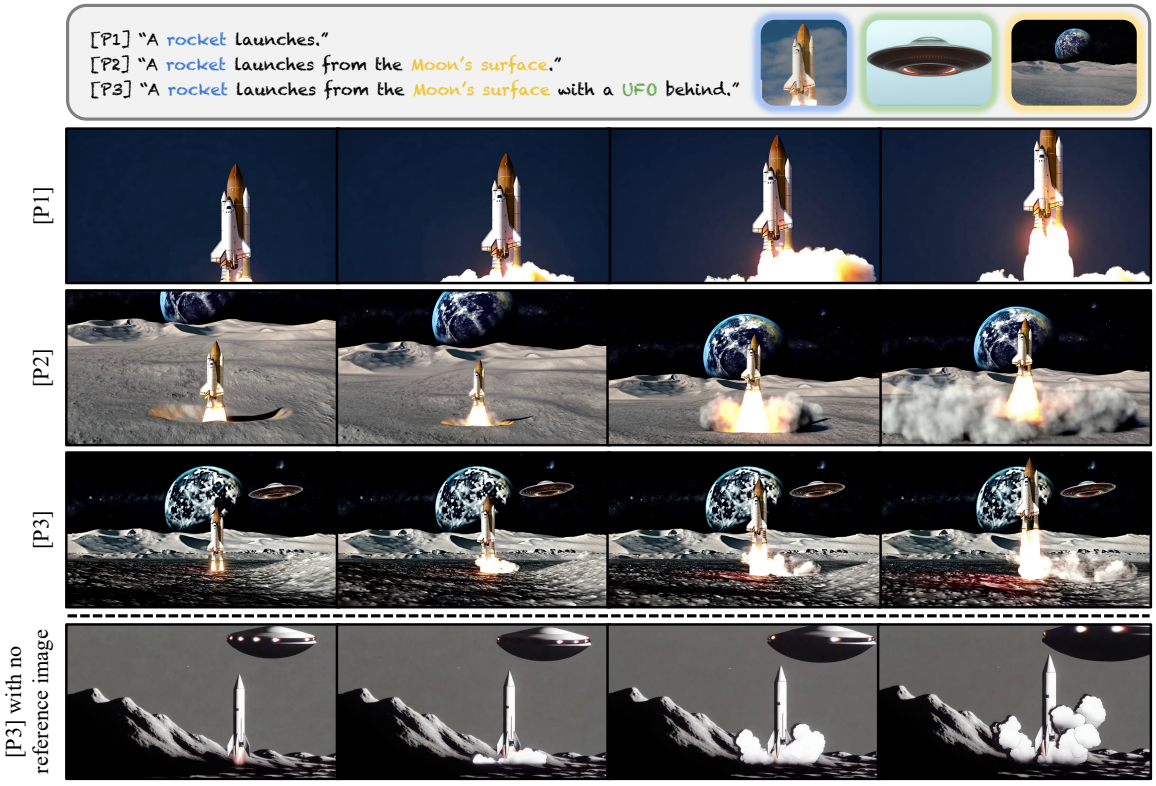

Figure 12. **Additional results of multi-subject open-set personalization.**

C.2. Same Text Prompt with Different Reference Images

Figure 1 presents videos generated using the same prompt and conditional subjects but varying background reference images. To demonstrate our model's robustness and ability to generate diverse visual content and motion, we showcase generated videos where the reference image for one subject is altered while keeping all other conditional inputs unchanged. Specifically, we provide samples with different reference images of *person* in Figure 13 and *dog* in Figure 14.

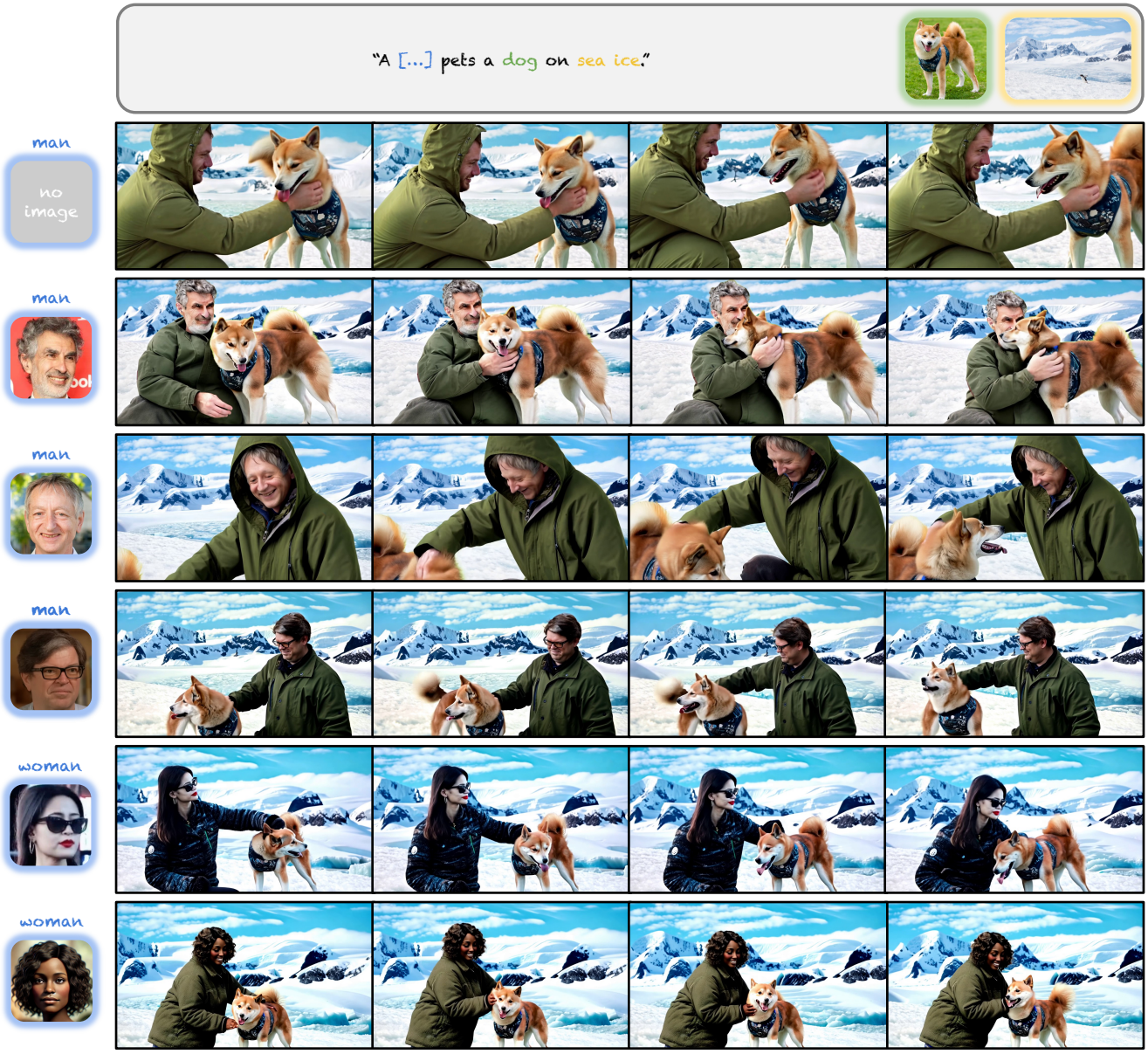

Figure 13. **Same text prompt with different reference images of *person*.**

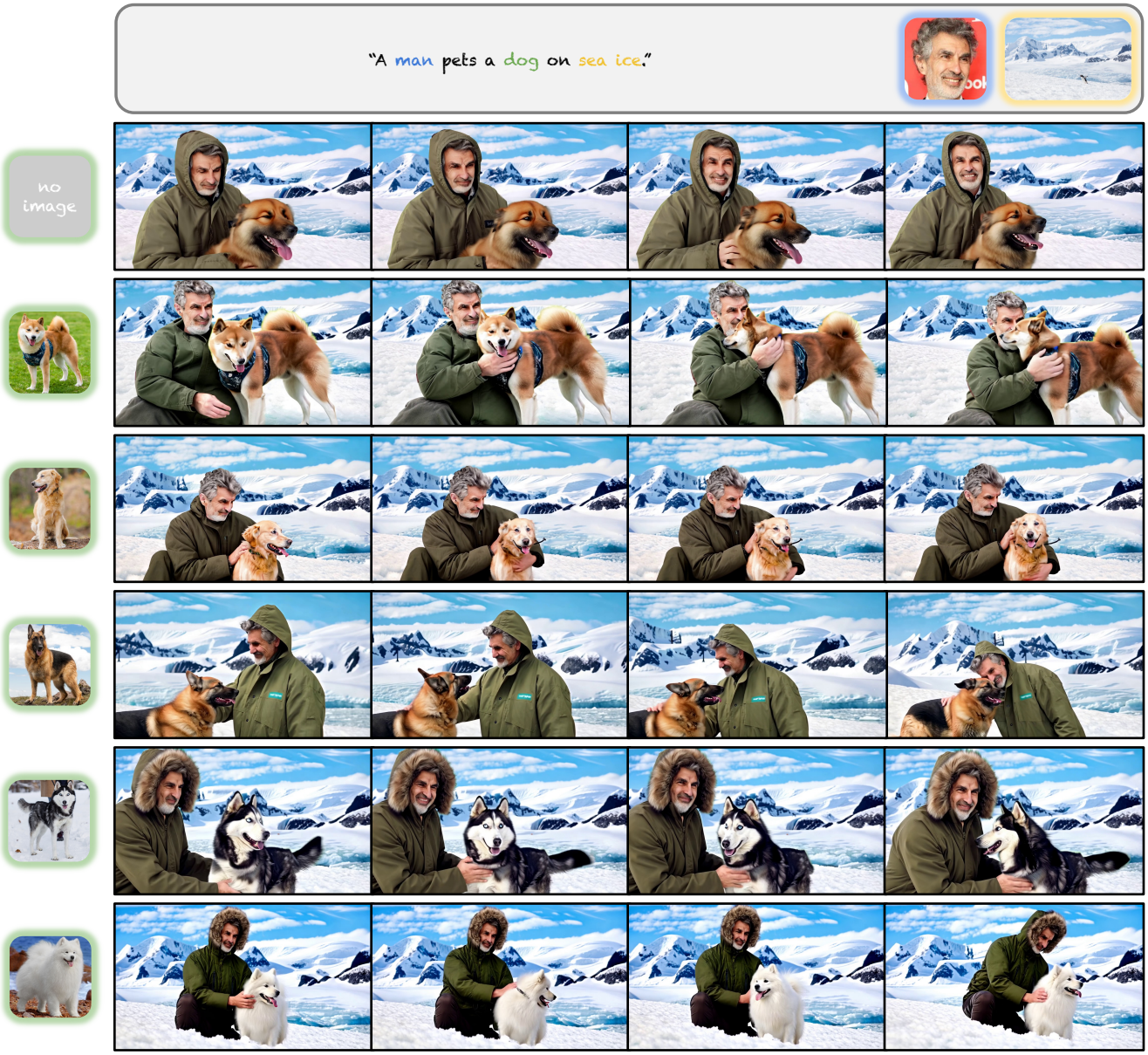

Figure 14. **Same text prompt with different reference images of *dog*.**

C.3. More Comparisons on Different Conditional Subjects

Figure 5 shows qualitative comparisons between *Video Alchemist* and state-of-the-art personalization models on the conditional subjects of *horse* and *woman*. In this section, we present more qualitative comparisons on other conditional subjects, including *dog* in Figure 15, *cat* in Figure 16, *car* in Figure 17, and *dinosaur toy* in Figure 18.

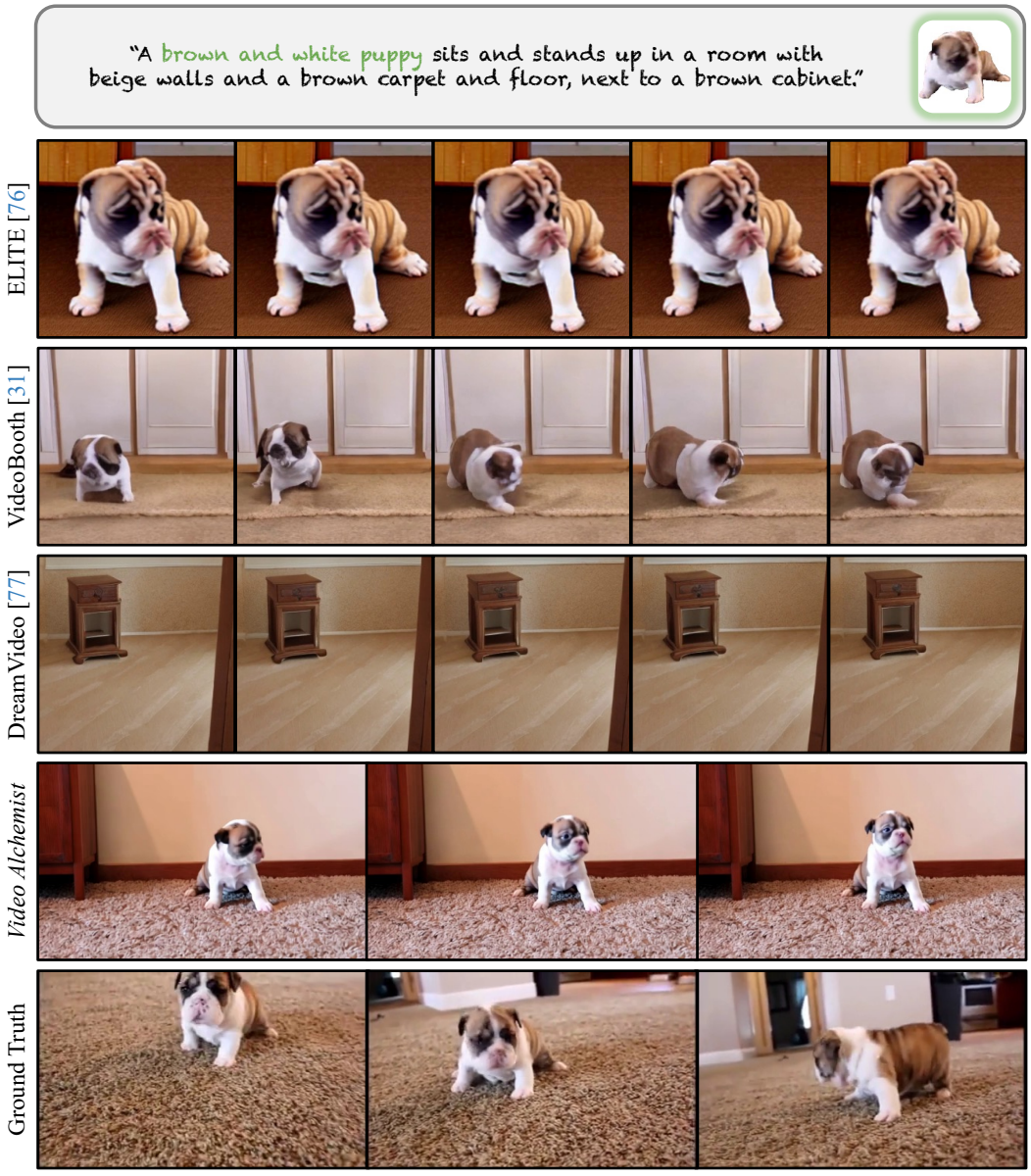

Figure 15. **Qualitative comparison on the conditional subject of *dog*.**

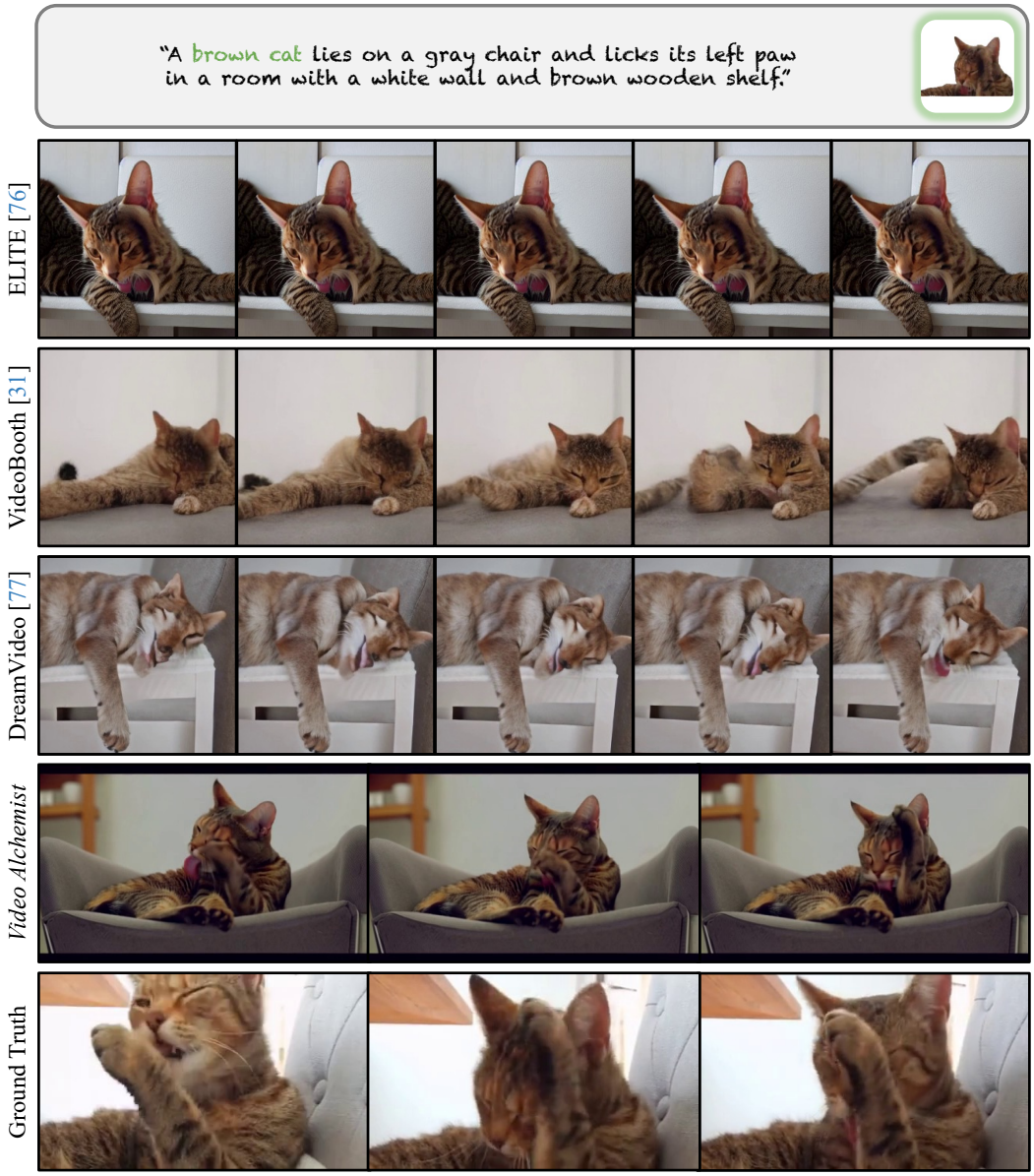

Figure 16. **Qualitative comparison on the conditional subject of *cat*.**

Figure 17. **Qualitative comparison on the conditional subject of *car*.**

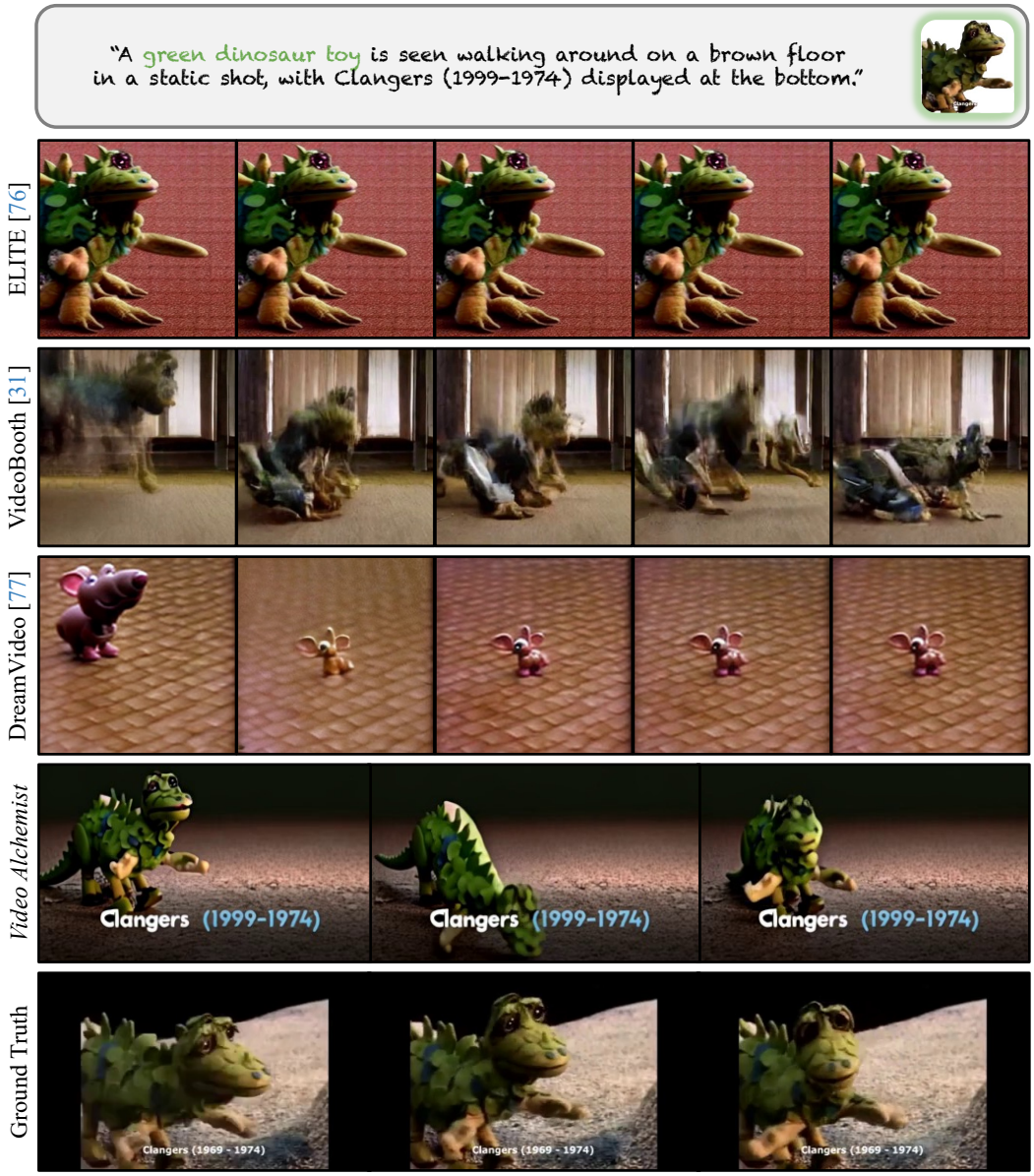

Figure 18. **Qualitative comparison on the conditional subject of *dinosaur toy*.**

D. Limitations

Model Overfitting. In Section 3.3 and Appendix A.3, we alleviate the model overfit by introducing data augmentation and random sampling with replacement during training. However, some undesirable image properties learned by the model remain unresolved. For example, *Video Alchemist* may sometimes generate subjects with facial expressions or postures similar to the reference images. Figure 5 shows that existing personalization models that adopt a similar reconstruction-based training, such as IP-Adapter [85], also exhibit the same problem, which remains a challenge for future work.

Taking Image Segments as Inputs. Our model personalizes video synthesis using segmented images as input. Thus, additional user efforts may be required if localization algorithms are unable to segment the intended subject accurately. To address this problem, we plan to include training samples in which the segmented images are pasted onto random background images to ease the need to segment the reference images.

Unnatural Composition for Multi-subject Conditioning. Empirically, for multi-subject conditioning, the synthetic videos sporadically exhibit unrealistic compositions and scales between different subjects. This behavior can be interpreted as the relative minority of videos with multiple subjects in the training dataset. We are considering creating a training dataset with a higher frequency of video samples with multiple subjects for future work.

Unsupported Measure on Video Quality. Like the CLIP similarity score [70], *MSRVTT-Personalization* does not assess visual quality. Users must rely on alternative evaluations, such as user studies, to compare visual quality.