Align Images Before You Generate

Supplementary Material

A. Training Details

The training strategy for MVAdapter+CorrAdapter* in Tables 1 and 2 is the same, referring to MVAdapter's original training settings [13]. The training dataset is a filtered subset of the Objaverse [4], comprising about 70k samples, captioned by Cap3D [29]. We render 6 orthographic views for each 3D object with elevation angle $0°$ and azimuth angles at $\{0°, 45°, 90°, 180°, 270°, 315°\}$ as supervision, and also render a random view within a certain frontal range to serve as an image condition. For both image- and text-conditioned generation, we choose MVAdapter-SDXL as the baseline. During training, we also randomly drop the text condition with a probability of 0.1, the image condition with a probability of 0.1, and both with a probability of 0.1. The log single-to-noise ratio (SNR) is set as $\log(n)$ where $n = 8$ as in MVAdapter's official codebase to keep the same noise level distribution as the pre-trained MVAdapter model. The LoRA [9] rank for finetuning is 16 with gradient clipping at 1.0. And we choose LoFTR [40] as the feature matcher with 0 matching threshold. For the specific training hyper-parameters, we set the batch size to 64 with gradient accumulation steps of 4, and the learning rate to 2×10^{-6} with warmup steps of 400. The model is fine-tuned for 1 epoch on 4 NVIDIA RTX 6000 Ada GPUs, taking about 24 hours. We also adopt DeepSpeed [34] with zero-3 offload to reduce the memory consumption.

B. More Baselines

To better support the plug-and-play claim and demonstrate the effectiveness of CorrAdapter, we add experiments on three more baselines, including the image-conditioned multi-view generation method Zero123++ [37], the text-conditioned multi-view generation method MVDream [39], and the text-conditioned video generation method HunyuanVideo [19]. We report the results of each baseline and its +CorrAdapter variant in Tables 5, 6, and 7, respectively, under the same protocols as in the main paper.

C. More Visual Results

We provide more visual results in this section as additional support of our main results.

C.1. Native Correspondences

We visualize more results of native correspondences and several matching baselines on generated images in Figure 6. Results prove that the native correspondences are reliable to guide the information interaction between aligned areas, even if under difficult situations like large viewpoint changes or sparse texture regions.

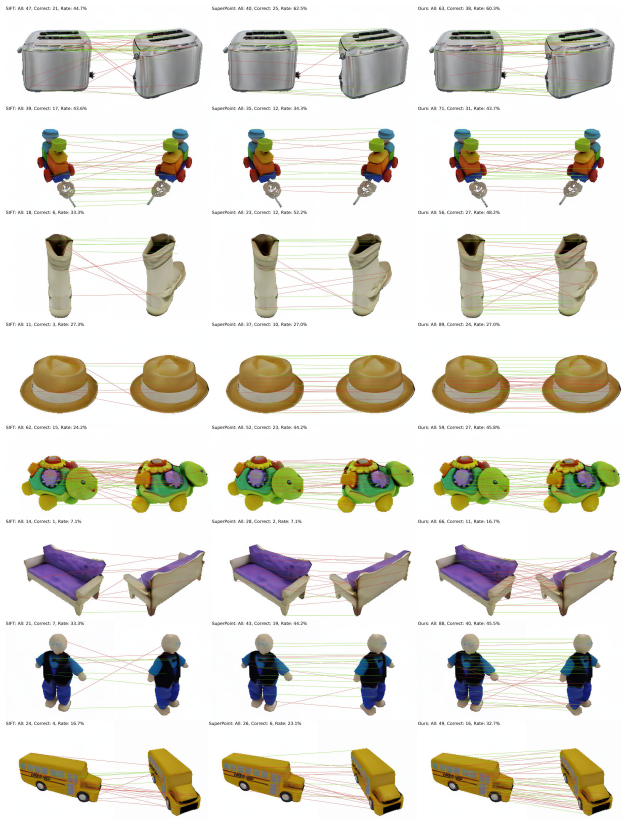

Figure 6. Visualizations of native correspondences and several matching baselines on generated images. The green lines indicate correct correspondences, and the red lines indicate incorrect ones. The native correspondences rival the matching baselines in terms of accuracy and matching number. Zoom in to see the details.

C.2. Static Multi-View Generation

The static multi-view generation is conducted on GSO dataset [6] for SyncDreamer [24], and on both GSO and Objaverse [4] datasets for MVAdapter [13]. Results are shown in Figures 7, 8, and 9. The superscript * indicates that the model is trained with our proposed training scheme.

C.3. Dynamic Video Generation

The dynamic video generation is conducted on the VBench dataset [12] for Wan2.1-1.3B (abbreviated as Wan2.1) [44]. Results are shown in Figure 10. Full videos can be found at https://github.com/SuhZhang/CorrAdapter.

Table 5. More results on image-conditioned multi-view generation.

Method	Single-Image Quality			Geometric Consistency					
	PSNR↑	SSIM↑	LPIPS↓	cPSNR↑	cSSIM↑	cLPIPS↓	CD↓	depth↓	MEt3R↓
Zero123++ [37]	20.70	0.8019	0.2129	19.27	0.7771	0.2589	10.56	78.46	0.4008
Zero123+++CorrAdapter	**21.55**	**0.8091**	**0.2117**	**19.66**	**0.7934**	**0.2501**	**10.54**	**77.94**	**0.3982**

Table 6. More results on text-conditioned multi-view generation.

Method	Single-Image Quality			Geometric Consistency					
	FID↓	IS↑	CLIP-Score↑	cPSNR↑	cSSIM↑	cLPIPS↓	CD↓	depth↓	MEt3R↓
MVDream [39]	32.06	11.10	33.06	15.86	0.6965	0.2670	**10.27**	70.72	0.2945
MVDream+CorrAdapter	**29.94**	**11.27**	**33.26**	**17.23**	**0.7474**	**0.2411**	10.31	**67.32**	**0.2728**

Table 7. More results on text-conditioned video generation. We generate 81 frames per video to match the Wan2.1-1.3B setup.

Method	Temporal Quality					Single-Frame Quality		Text-Video Consistency		
	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Scene	Appearance Style	Overall Consistency
HunyuanVideo [19]	0.9659	0.9761	0.9949	0.9932	**0.5694**	**0.6296**	0.6686	**0.3234**	0.1874	0.2503
HunyuanVideo+CorrAdapter	**0.9732**	**0.9815**	**0.9963**	**0.9938**	0.5278	0.6292	**0.6883**	0.3075	**0.1879**	**0.2571**

D. Limitations and Future Work

The CorrAdapter proposed in this paper is designed to enhance the spatiotemporal consistency of generated images, thus the baseline-inherited inconsistency between outputs and conditions may still exist in some particular cases, as shown in Figure 11, even though the consistency between generated images are increased after CorrAdapter is applied. To enhance the perception ability for multi-modal conditions (text, image, *etc.*), further research will explore the underlying cues in the diffusion process to align the outputs with conditions better.

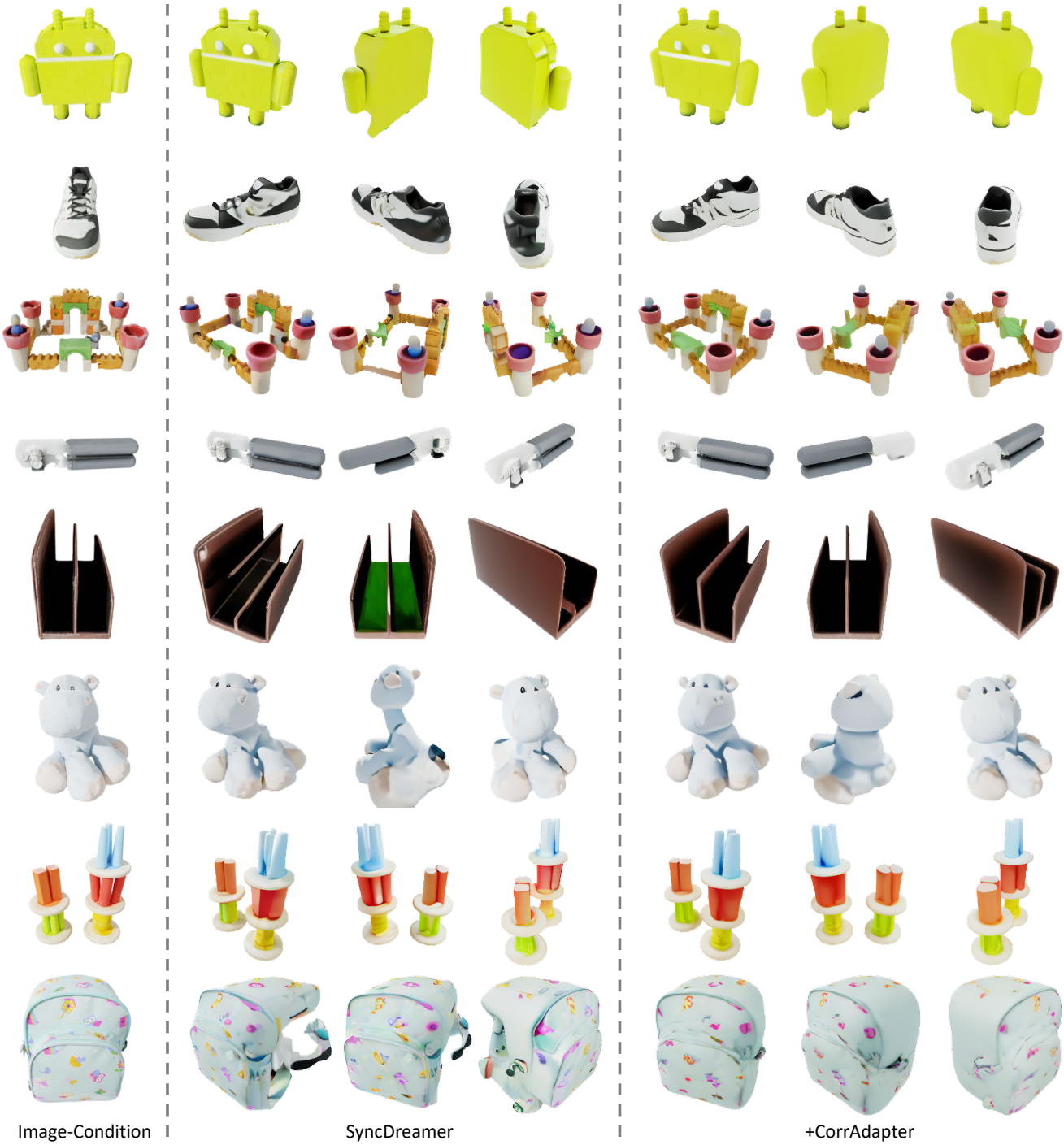

Figure 7. Visualizations of CorrAdapter on SyncDreamer for image-conditioned multi-view generation on GSO dataset.

Figure 8. Visualizations of CorrAdapter on MVAdapter for image-conditioned multi-view generation on GSO and Objaverse datasets.

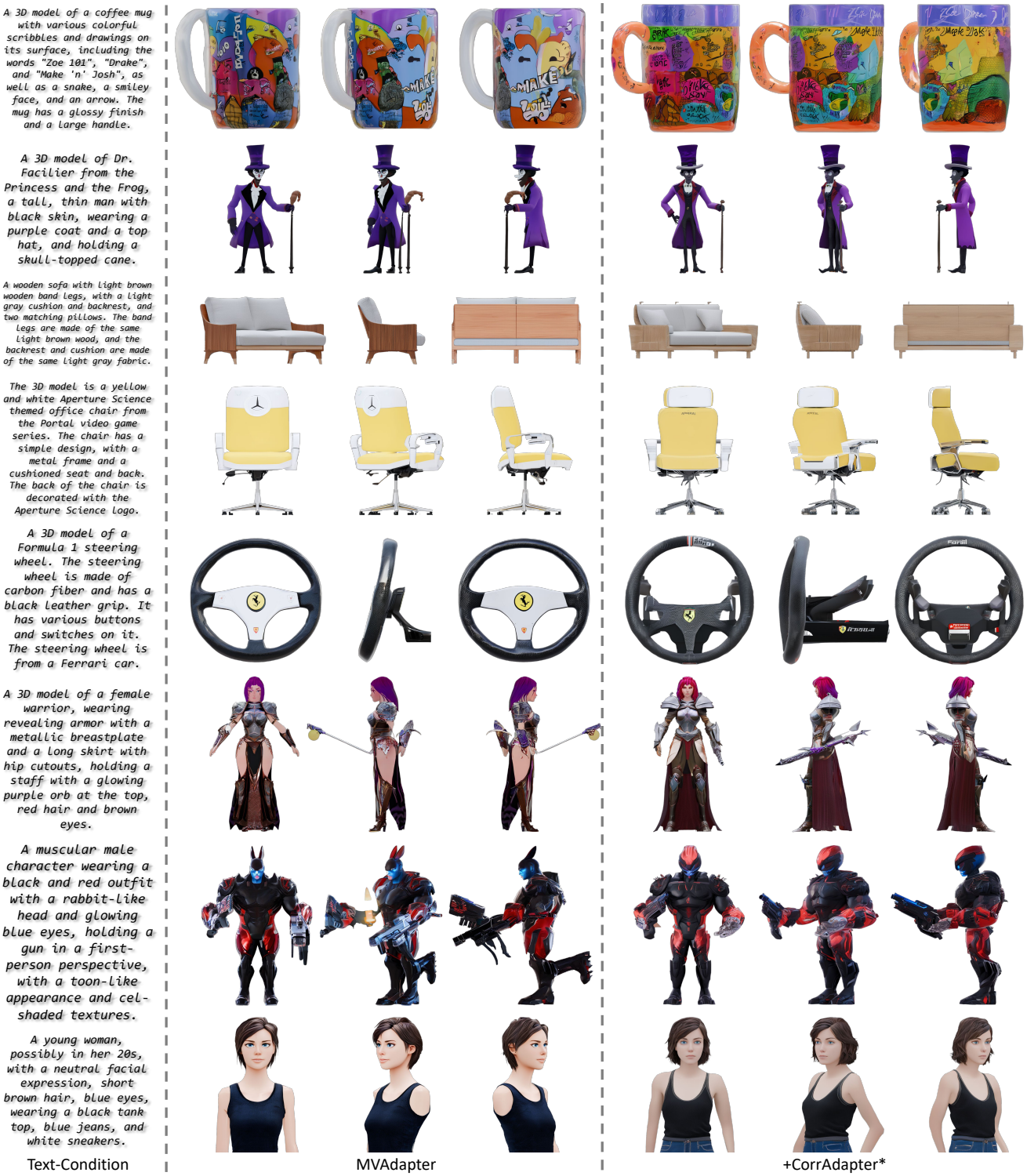

Figure 9. Visualizations of CorrAdapter on MVAdapter for text-conditioned multi-view generation on Objaverse dataset.

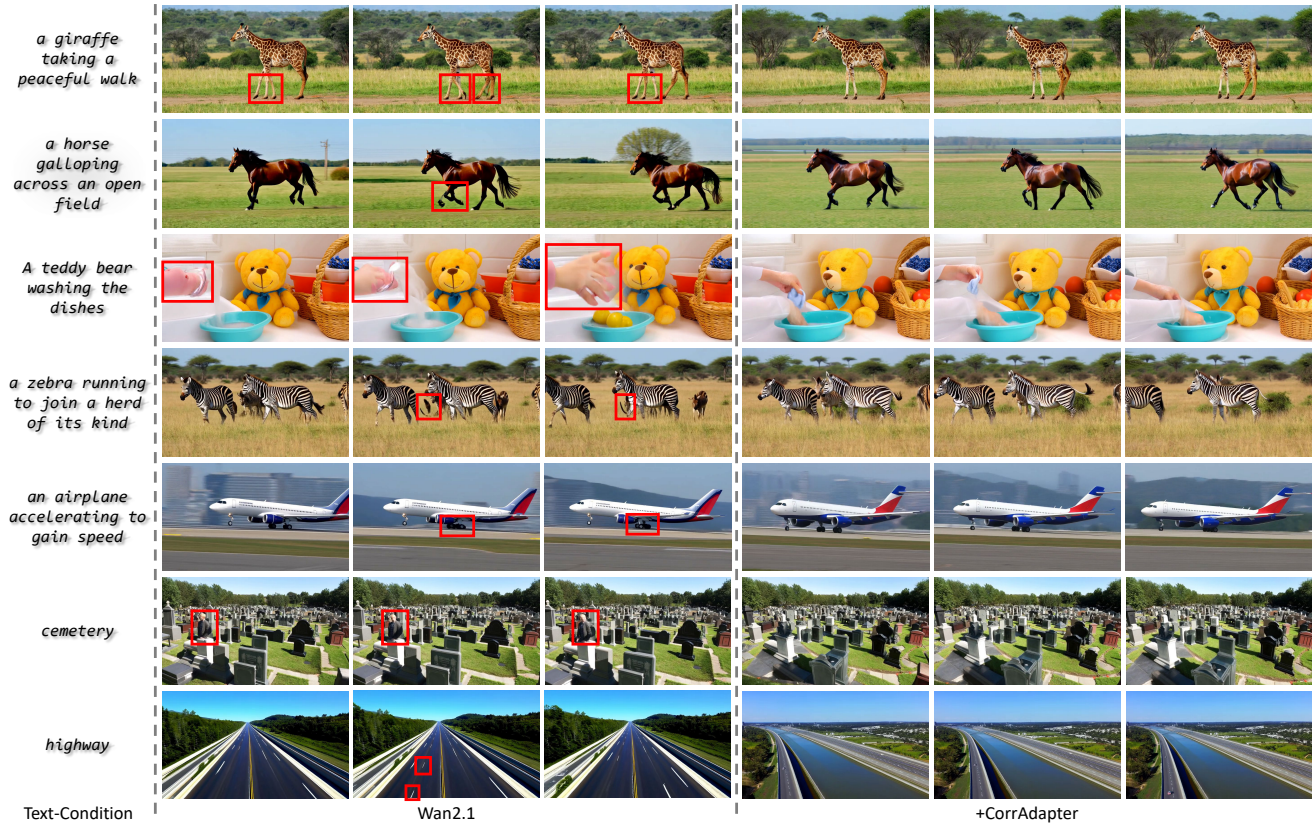

Figure 10. Visualizations of CorrAdapter on Wan2.1 for text-conditioned video generation on VBench dataset. We use red boxes to highlight the broken, distorted, or chaotic regions in the original generated images of Wan2.1 for comparison. Zoom in to see the details.

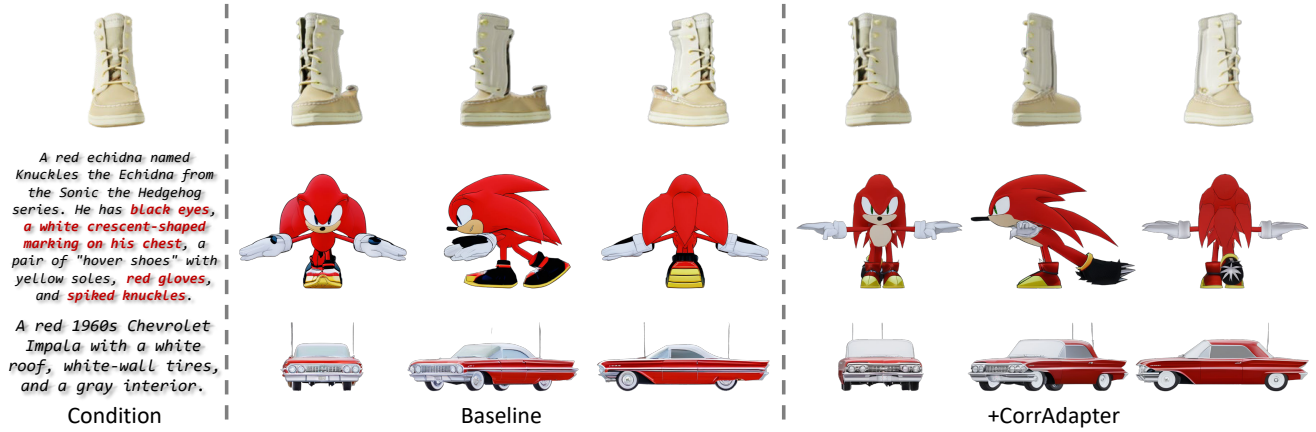

Figure 11. Visualizations of the baseline-inherited inconsistency between generated outputs and conditions in some particular cases. For the first row, the model cannot correctly fit the unseen back side. For the second row, the generated images do not match the red text. For the third row, the position of the car antenna is not logical.