A. Implementation Details

A.1. Model Checkpoints

We use the pre-trained T2V diffusion model LaVie and VideoCrafter2, available at https://github.com/Vchitect/LaVie and https://github.com/AILab-CVC/VideoCrafter, respectively. For LaVie, the Stable Diffusion v1.4 model is employed to encode and decode latent. We also utilize CLIP from https://huggingface.co/openai/clip-vit-base-patch32 and the ImageReward model from https://github.com/THUDM/ImageReward.

A.2. Evaluation Details

During the video guidance process, we extract key frames from the video—specifically, the first, sixth, eleventh, and sixteenth frames—and assess the reward. When using an LVLM as the reward model, we concatenate the key frames using the following scripts:

```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
key_frames = [0, 5, 10, 15]

for idx, frame in enumerate(key_frames):
    ax = axes[idx // 2, idx % 2]
    ax.imshow(video[0, :, frame, :, :].permute(1, 2, 0).cpu().numpy())
    ax.axis('off')
    ax.set_title(f'Frame {frame + 1}')

# Adjust the layout and show the plot
plt.tight_layout()
plt.savefig(f'frame_{i}_{j}.png')
```

Listing 1. Pseudo-code for stitching key frames at once.

Next, we provide a system instruction that allows the LVLM to understand the sequence order and explicitly describes the task it should perform.

```
You are a useful helper that responds to video quality assessments.
The given image is a grid of four key frames of a video: the top left is the first frame, the top right is the second frame, the bottom left is the third frame, and finally the bottom right is the fourth frame.
Answer the reason first and the final answer later. Start the reason first with 'Reasoning: ' in front of the reason part and review your reasoning logically.
After reviewing your reasoning, give the final answer with 'Answer: '.
You should check all frame and comparing them, and ensure your reasoning leads to a sound final answer.
Your final 'answer' should one score only and the score must be from 1 to 9 without decimals.
Let's think step by step.
```

Listing 2. System instruction for GPT-4o

For a given video, we input the user prompt to the LVLM as follows:

```
For a given image as keyframes of video, Rate the following questions:
Considering all four images, does the prompt, prompt, describe the video well enough?
Review your reasoning thoroughly and then respond with your final decision prefixed by Answer: '.
```

Listing 3. User prompt for GPT-4o

where **`prompt`** is the given text prompt (*e.g.* "a bird and a cat")

B. Limitation

Sampling in our approach requires additional processing time to approximate the gradient. While our approach extends sampling time compared to baseline, it uniquely enables guidance with non-differentiable reward models such as LVLM APIs. Additionally, the effectiveness of our framework is influenced by the accuracy of the reward function, which opens avenues for further improvements as reward models continue to advance.

C. Additional Ablation Study

Number of Samples We analyze the effect of the sampling quantity on text alignment performance, evaluating the average text alignment score using the LaVie model with a CLIP reward model. As shown in Table 7, we find an optimal sampling size at $n = 5$. Increasing the number of samples increases the likelihood of selecting a denoised video that aligns with the desired control. However, excessive sampling introduces a risk: errors predicted by Tweedie's formula in initial sampling steps may result in irreversible changes, affecting video quality negatively.

n	Avg.
1	0.3722
3	0.3749
5	**0.3780**
10	0.3705

Table 7. Quantitative results on text alignment by sample size.

Guidance Range We also evaluate the effect of the guidance range with the same baseline. Table 8 reveals that applying guidance in the early stages is more effective than in later stages, as these initial steps establish the overall spatial structure of the video. However, extending the guidance range too far allows errors in the approximated optimal control to accumulate, ultimately degrading the quality of the final output video.

Guidance Step	Avg.
None	0.3722
$t \in [T, T-5]$	**0.3780**
$t \in [T-5, T-10]$	0.3769
$t \in [T, T-10]$	0.3635

Table 8. Quantitative results on text alignment by range of guidance step.

Assessment policy using LVLM We evaluate the impact of the assessment protocol in LVLM by analyzing the average scores generated with the VideoCrafter2 model. Specifically, we modify the system prompt to instruct LVLM to answer only with 'yes' or 'no' when assessing text-video alignment. The alignment score is then derived by calculating the percentage of the top 5 logits that correspond to 'yes'. Table 9 reveals that scoring alignment on a scale from 1 to 9 achieves better performance in terms of text alignment. This is likely because a broader scale allows for more nuanced distinctions in fidelity, enabling LVLM to capture subtle differences in text-video alignment more effectively.

Method	Text Alignment	General Quailty	Avg.
VC2	0.4129	0.7617	0.5873
+GPT$_{0/1}$	0.4358	**0.7550**	0.5954
+GPT$_{1\text{-}9}$	**0.4425**	0.7537	**0.5981**

Table 9. Average results by assessment policy using LVLM.

D. Additional Analysis

Method	Appearance Style	Temporal Style	Human Action	Multiple Objects	Spatial Relationship	Overall Consistency	**Avg.**
LaVie	0.2312	0.2502	0.9300	0.2027	0.3496	0.2694	0.3722
+ GPT4o	0.2366	**0.2508**	0.9300	**0.2546**	**0.3531**	0.2709	**0.3827**
+ Qwen2.5-VL 3B Image	**0.2388**	0.2447	**0.9700**	0.2477	0.3238	0.2647	0.3816
+ Qwen2.5-VL 3B Video	0.2325	0.2464	**0.9700**	0.2431	0.3101	**0.2738**	0.3793
LTX-Video-2B	0.2189	0.1784	**0.5303**	0.1994	0.3436	0.1916	0.2770
+ GPT4o	**0.2202**	**0.1813**	0.5051	**0.2335**	**0.4177**	**0.1947**	**0.2921**

Table 10. Baseline comparison with open-source Image and Video LVLM and longer video generation model.

Aspects	**Baselines**	**Ours**
Overall Quality	2.61	3.19
Temporal Quality	2.65	3.21
Text Alignment	2.60	3.94

Table 11. User study.

Method	GPU Memory	Computing Time
Lavie	4.4 GiB	22.7 s/video
+Ours	7.5 GiB	154.5 s/video

Table 12. Computation.

Open-source LVLM. We leverage an open-source LVLM (QWen2.5-VL 3B) using both stitched image input and direct video input. As shown in Table 10, our framework consistently improves T2V alignment. Interestingly, image input demonstrated stronger performance than direct video input for this specific LVLM. We hypothesize this might be due to our frame stitching method effectively highlighting key temporal information for the LVLM.

Method	Style		Semantics			Condition Consistency	Avg.
	Appearance Style	Temporal Style	Human Action	Multiple Objects	Spatial Relationship	Overall Consistency	
LaVie [43]	0.2312	0.2502	0.9300	0.2027	0.3496	0.2694	0.3722
+ CLIP	0.2370 (+2.5%)	0.2490 (-0.5%)	0.9400 (+1.1%)	0.2607 (+28.6%)	0.3074 (-12.1%)	0.2738 (+1.6%)	0.3780
+ ViCLIP	0.2348 (+1.6%)	0.2485 (-0.7%)	0.9600 (+3.2%)	0.2149 (+6.0%)	0.2872 (-17.9%)	0.2752 (+2.1%)	0.3701
+ GPT	0.2366 (+2.3%)	0.2508 (+0.2%)	0.9300 (-0.0%)	0.2546 (+25.6%)	0.3531 (+1.0%)	0.2709 (+0.6%)	0.3827

Method	Temporal Consistency		Dynamics		Frame-wise Quality		Avg.
	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	
LaVie [43]	0.9450	0.9689	0.9718	0.4799	0.5687	0.6611	0.7659
+ CLIP	0.9495 (+0.5%)	0.9712 (+0.2%)	0.9735 (+0.2%)	0.4560 (-5.0%)	0.5727 (0.7%)	0.6637 (+0.4%)	0.7644
+ ViCLIP	0.9443 (-0.1%)	0.9694 (+0.0%)	0.9741 (+0.2%)	0.4707 (-1.9%)	0.5746 (1.0%)	0.6487 (-1.9%)	0.7636
+ GPT	0.9470 (+0.2%)	0.9693 (+0.0%)	0.9742 (+0.2%)	0.4725 (-1.5%)	0.5726 (+0.7%)	0.6615 (+0.1%)	0.7662

Table 13. Comparison with video-based reward model. Higher numbers indicate better video quality. The numbers in parentheses denote the performance difference from the baselines.

Long Video Generation Model. To address concerns about generalization to longer videos, we applied Free^{2}Guide to a long video generation model (LTX-video 2B), generating 15-second videos. As presented in Table 10, we measure VBench2-beta-long metrics and our framework significantly improves performance over the baseline (which used stochastic sampling for fair comparison), demonstrating its effectiveness in longer videos.

User Study. We conducted a user study with 50 participants on Prolific, comparing videos from our method against the baseline (LaVie and VideoCrafter2). Participants rated videos on a 1-5 scale for overall quality, temporal quality, and text alignment. Our method was consistently preferred across all aspects, as shown in Table 11.

D.1. Video Reward Guidance

While using a video-based reward model to guide videos is a more natural approach, we claim that video reward models fail to capture the representation needed for guidance because the dataset of video-text pairs is relatively limited compared to images. To support this, we compare the results of using a video-based reward model for guidance with a video-based reward model for text alignment. We adopt ViCLIP [44], a pre-trained video-text representation learning model available at `https://huggingface.co/OpenGVLab/ViCLIP`, as the video reward model. Using LaVie as the baseline, we compute the reward based on eight video frames, measuring the similarity between the video and text embeddings.

Table 13 shows that the video-based reward model does not significantly outperform the image-based reward model. However, it specifically enhances the Overall Consistency and Dynamic Degree metrics. It is worth noting that the Overall Consistency metric is evaluated using ViCLIP itself, which could introduce a bias favoring the video reward model. In addition, we observe that ViCLIP struggles with spatial information processing compared to CLIP, leading to lower performance on the Multiple Objects and Spatial Relationship metrics. These results highlight the challenges of video reward models to fully capture the relationship between video and text due to the lack of training datasets.

D.2. Video Inverse Problems

Our framework can readily extend to inverse problems in the video domain, building on approaches from previous work [17, 51]. In Figure 5, we show a video reconstructed by our method using $\times 16$ average pooling on spatial resolution. For the reward function, we use the L_2 distance between the corrupted denoised video and the corrupted video, applying a sampling size of 10 at each step with DDIM over 500 steps, using VideoCrafter2. Our results demonstrate that, compared to unguided sampling, our method generates realistic videos that remain faithful to the input. We leave further extension to video inverse problems as future work.

E. Additional Visual Results

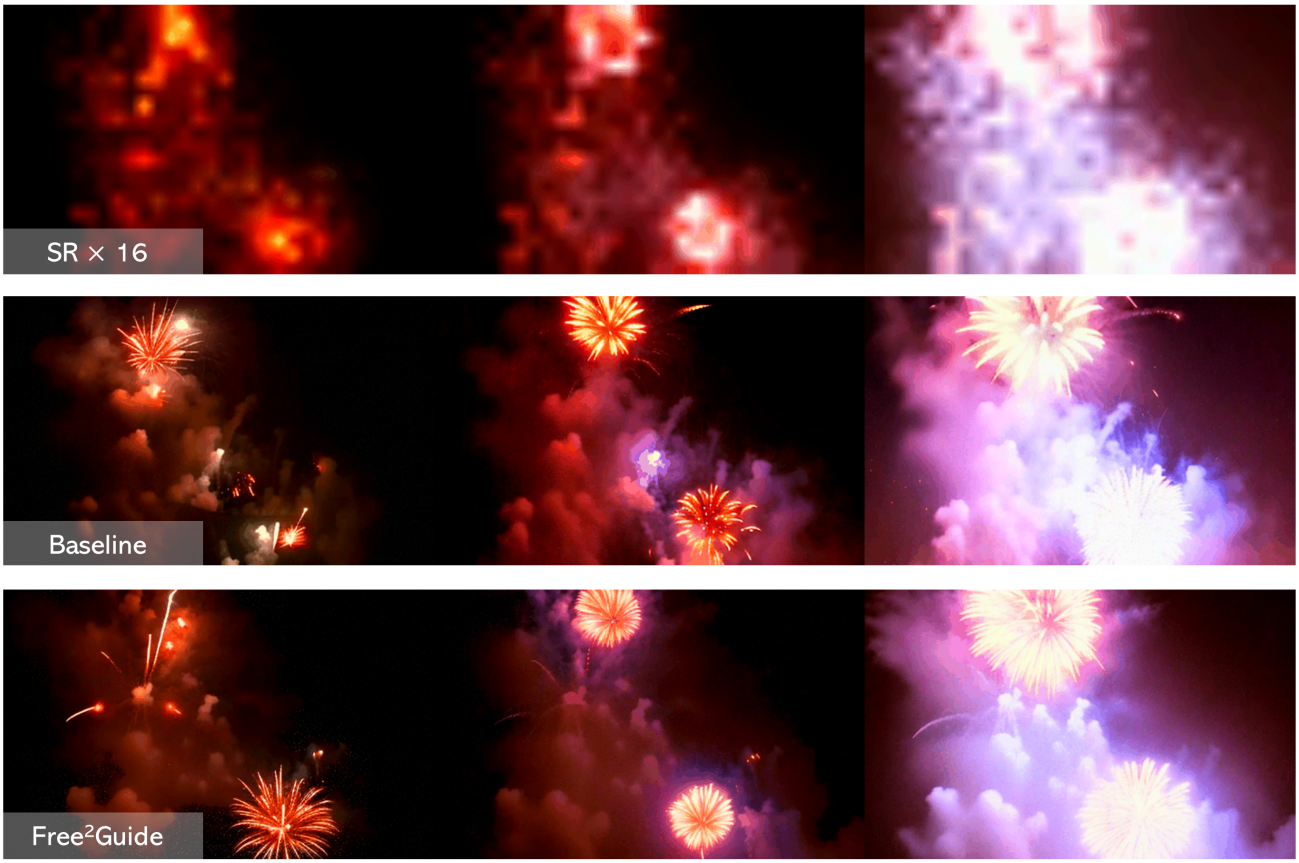

Figure 5. The result of applying our method to the inverse problem. Baseline represents that no guidance is applied during sampling.

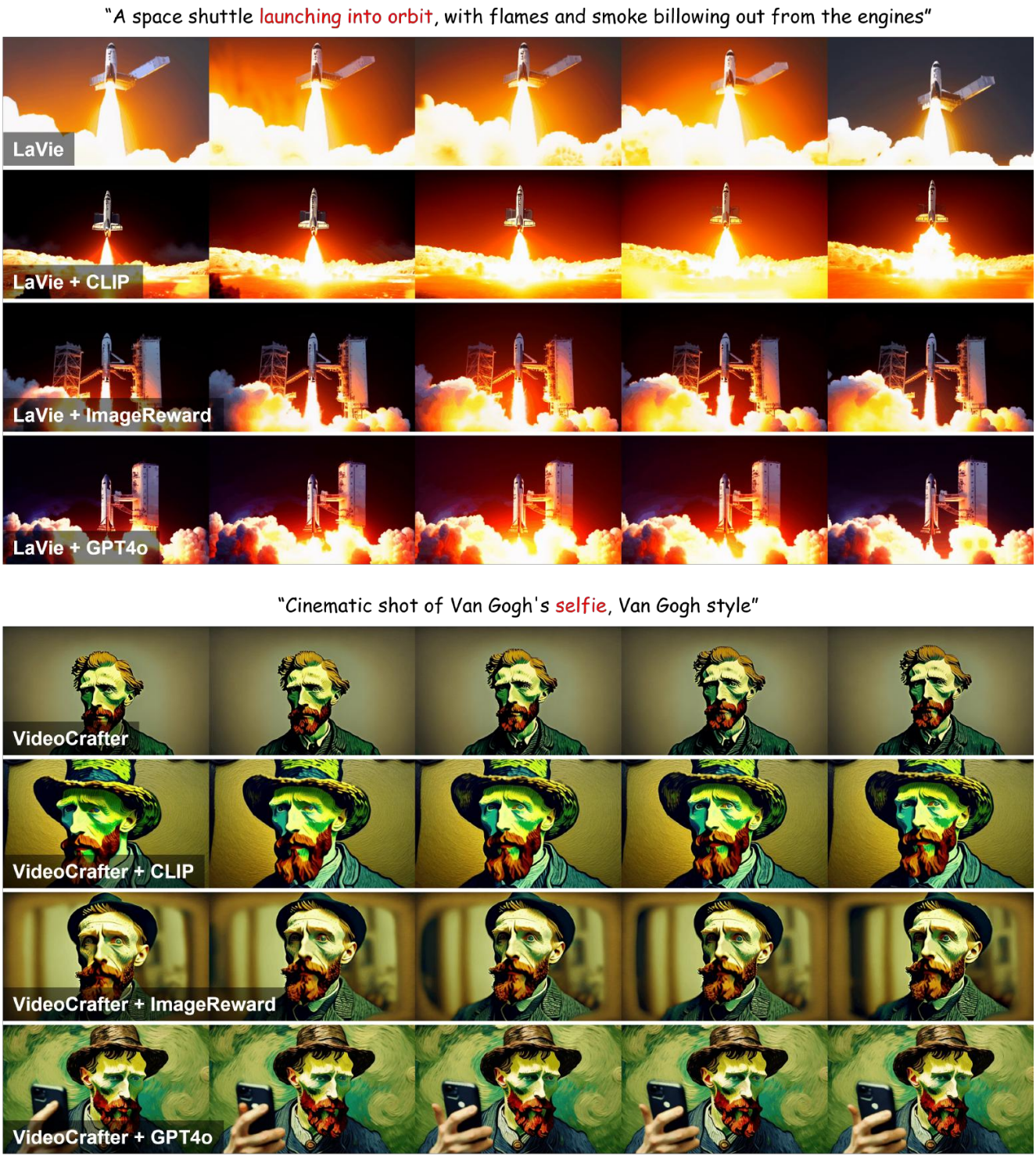

Figure 6. More qualitative comparison of different reward models. The red text highlights the difference between the models.

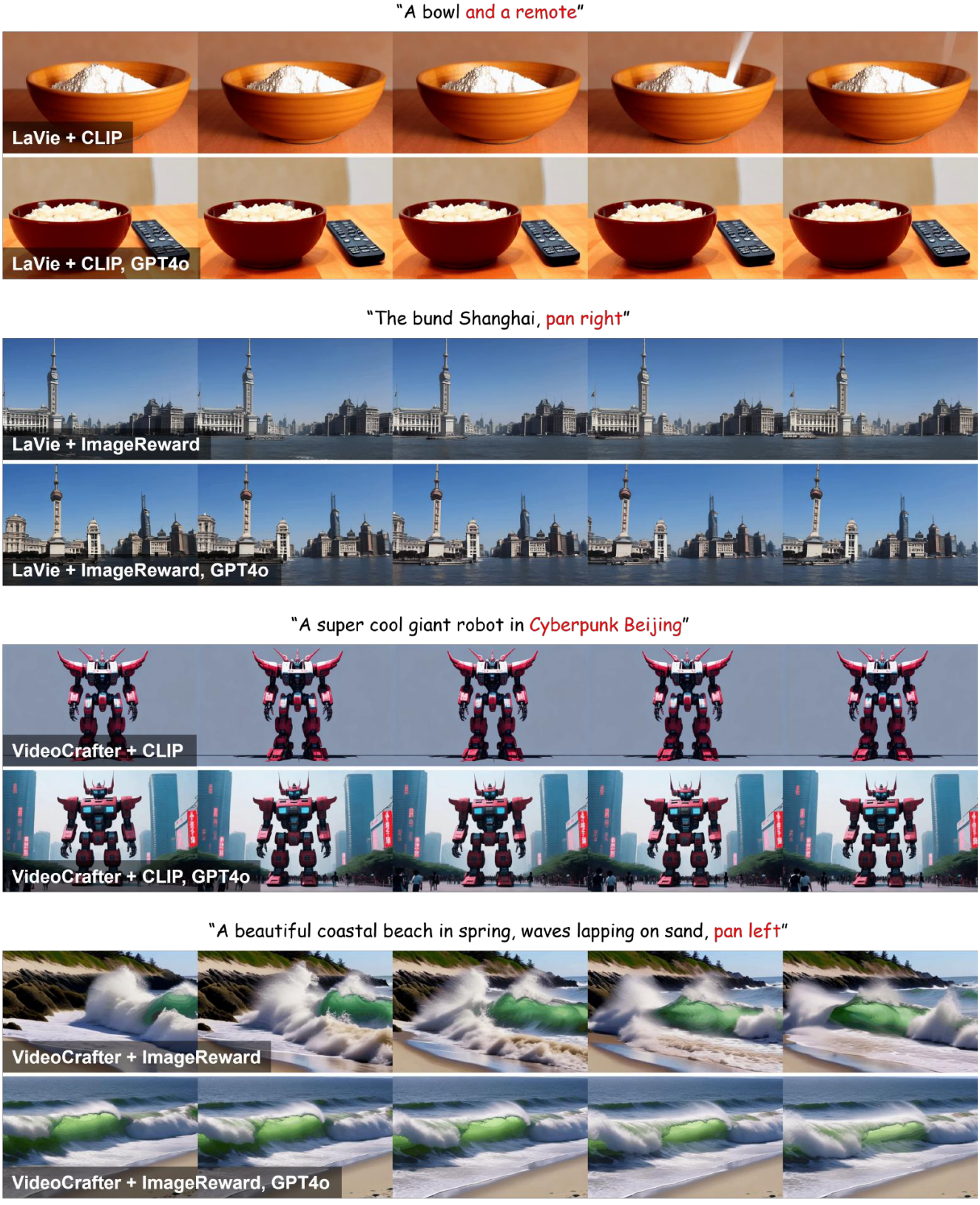

Figure 7. More qualitative results of ensembling with LVLMs. The red text highlights the difference between the models.

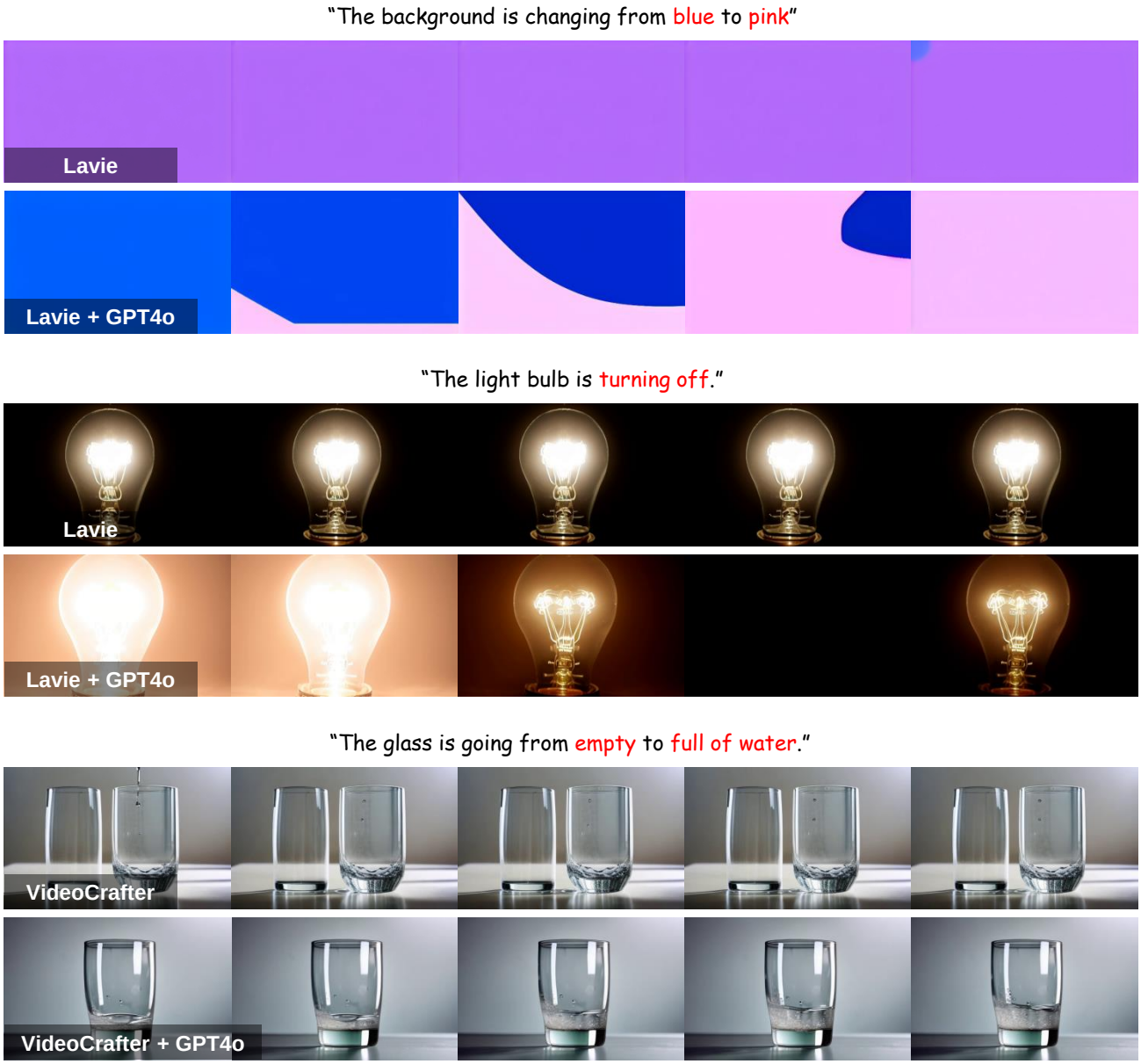

Figure 8. More qualitative comparison of T2V-Compbench to analyze video-specific dynamics. The red text highlights the difference between the models.