Optical-Flow Guided Prompt Optimization for Coherent Video Generation

Supplementary Material

The supplementary sections are organized as follows. Suppl Sec. A delves into the details on our training, inference configurations and evaluation setups. Suppl Sec. B presents additional experimental analyses. Finally, Suppl Sec. C provides further results obtained from our framework.

A. Implementation and Evaluation Details

A.1. Training Details of the Discriminator

The discriminator was trained to distinguish between real and generated optical flow representations, aiming to enhance temporal consistency in video generation. For training, we use a batch size of 32 and the SGD optimizer with a learning rate of 0.0005 and a momentum of 0.9. The training process spans approximately 20 epochs, with the model parameters from the epoch achieving the best validation loss being selected.

A.2. Hyperparameters for Evaluation

Tab. 5 lists the hyperparameters used for our quantitative evaluation. The '# of frames' denotes the number of video frames decoded into pixels and used for optical flow calculation. Specifically, if the value is 6, this indicates that 3 sets of adjacent frames were sampled, resulting in 3 optical flows being used to compute the corresponding loss.

	Lavie	AnimateDiff	VideoCrafter2
# of tokens	1	1	1
K (opt iter)	3	3	3
opt range	$5<t<15$	$3<t<15$	$3<t<20$
# of frames	6	2	6
λ_1	1.0	1.0	1.0
λ_2	1.0	5.0	5.0
λ_3	10.0	10.0	3.0
η (lr)	0.0005	0.005	0.001

Table 5. Evaluation hyperparameters used for each model.

A.3. User study

We conduct an A/B user study on Prolific, where 100 participants compared total 90 pairs of videos. Each participant was asked to choose the video they found more natural and smooth in motion by asking "Choose the more natural and smoothly moving video.". The answer options were: A, B, or Both.

B. Additional Analysis

In this section, we present additional analyses to evaluate the generalization capability and performance of our approach across various scenarios. The experiments primarily focus on AnimateDiff [14].

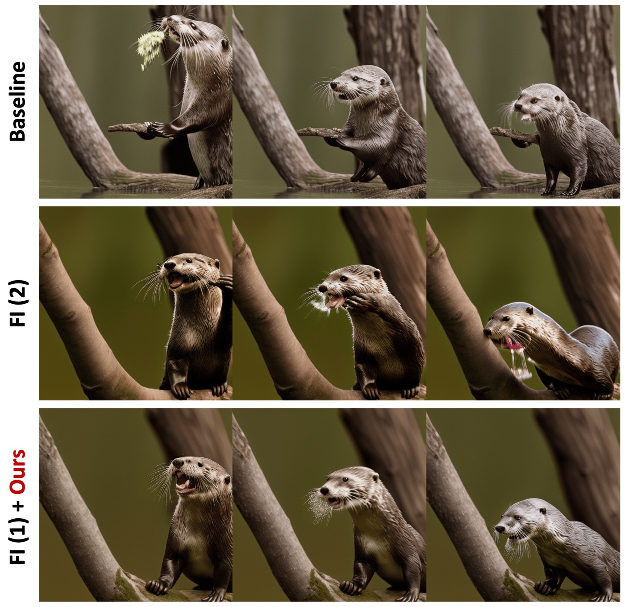

Figure 6. Qualitative comparison between the baseline, FreeInit, and FreeInit combined with our method. When FreeInit is used repeatedly, videos tend to lose detail and exhibit saturation issues. In contrast, combining our method with a single application of FreeInit mitigates these problems while improving temporal quality.

B.1. Ablation on Token Optimization

Here, we conduct an ablation study to investigate the impact of various configurations related to token optimization on the overall performance (Tab. 6). Except for the factors being examined, all other settings were kept consistent with the values in Tab. 5.

The number of tokens First, we examine whether increasing the number of tokens enhances the optimization effect. Specifically, we added three additional tokens initialized as "authentic", "real" and "clear". Although this increased the factors that could potentially improve temporal quality, leading to an improvement in some temporal quality metrics, it also resulted in a decline in dynamic degree and overall consistency. Consequently, we determined that the benefits were not significant enough and chose to use a single token as the default setting.

The placement of tokens In addition, we investigate the effect of the learnable token's placement. Instead of append-

Method	Temporal Quality					Text Alignment
	Subject Consistency (↑)	Background Consistency (↑)	Temporal Flickering (↑)	Motion Smoothness (↑)	Dynamic Degree (↑)	Overall Consistency (↑)
Baseline	0.9488	0.9755	0.9228	0.9578	0.4700	0.2532
Baseline+ **Ours**	0.9528	0.9763	0.9258	0.9599	0.4125	0.2529
Increased Tokens (3)	0.9530	0.9760	0.9259	0.9605	0.3938	0.2498
Tokens Placed at Front	0.9507	0.9730	0.9244	0.9615	0.3730	0.2423
Init with 'the'	0.9509	0.9760	0.9263	0.9599	0.4438	0.2527

Table 6. Ablation results comparing the baseline, default setting, increased token count (3 tokens), tokens placed at the front, and tokens initialized with the word 'the'. Evaluation metrics are reported for subject consistency, background consistency, temporal flickering, motion smoothness, dynamic degree, and overall consistency.

	FI(1) + **Ours**	FI(2)	FI(4)
Subject Consistency	0.9669	0.9662	**0.9711**
Background Consistency	0.9834	0.9828	**0.9854**
Temporal Flickering	0.9579	0.9571	**0.9672**
Motion Smoothness	0.9776	0.9771	**0.9823**
Dynamic Degree	0.2850	**0.2988**	0.2600
Overall Consistency	**0.2469**	0.2463	0.2424
Image Quality	**0.6768**	0.6756	0.6435

Table 7. Quantitative results of FreeInit and FreeInit combined with our method. FI denotes FreeInit, and the number in parentheses indicates the number of noise initialization steps performed. **Bold**: Best, Underline: Second Best.

ing the learnable token to the end of the given prompt, we placed it at the front of the prompt to evaluate its impact. Similar to the observations in Um and Ye [35], we also find that appending the token to the end of the given prompt is more effective. This aligns with the general practice of structuring sentences where content-related information is stated first, followed by descriptive elements like adjectives.

Robustness to Initialization Words In main paper, we demonstrate that the cosine similarity with the initialization token starts sufficiently low and gradually decreases over time, indicating that the improvement is not merely a result of adding tokens. To further support this, we initialize the token with a seemingly unrelated word, 'the', and assess its impact on video generation quality. While the performance improvement was smaller compared to our final configuration, it still outperformed the baseline. This further reinforces the effectiveness and robustness of our method.

B.2. Synergies with the Existing Method

We demonstrate how our approach can be combined with orthogonal methods to achieve enhanced performance. For instance, while FreeInit [39] focuses on initializing noise, our method emphasizes guidance through prompt optimization. These complementary mechanisms allow the two approaches to work synergistically. In this section, we present the results obtained by using both methods together, highlighting their combined potential.

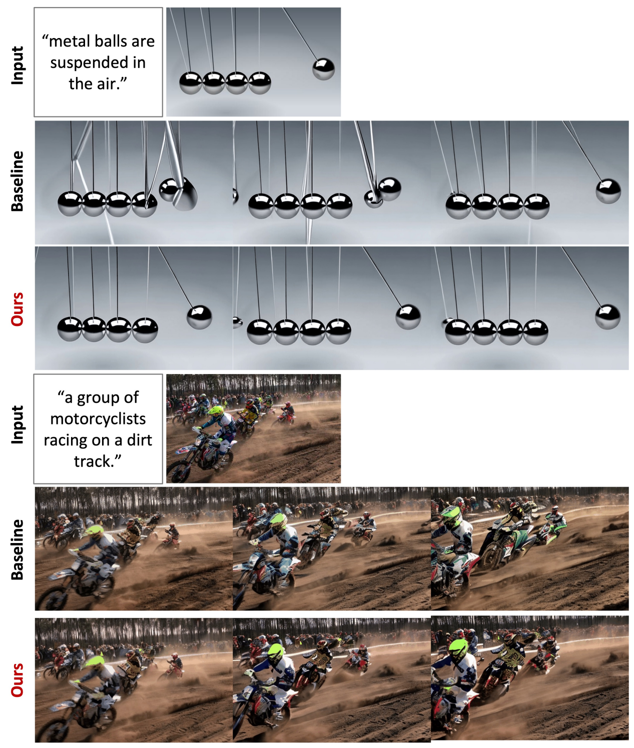
Input
"metal balls are suspended in the air."
Baseline
Ours
Input
"a group of motorcyclists racing on a dirt track."
Baseline
Ours

Figure 7. Additional results of DynamicCrafter [40].

While FreeInit significantly improves temporal quality, it does so at the expense of overall video quality. Specifically, Tab. 7 demonstrates that increasing the number of initialization steps leads to a sharp decline in metrics related to video generation quality, such as Overall Consistency and Image Quality. Fig. 6 further reveal that, compared to the baseline, the videos generated with FreeInit often lose high-frequency details and exhibit noticeable saturation. However, by applying our method after a single noise initialization step, we observed an improvement in temporal quality while relatively minimizing the compromise in video quality, compared to the FreeInit approach where noise initialization is applied multiple times.

B.3. Computational Cost

We compute GPU memory usage and computing time, averaging over 100 generations.

Model	GPU Memory	Computing Time
AnimateDiff	11.38 GiB	18.94 s/video
+Ours	35.60 GiB	50.94 s/video

Figure 8. Computational cost.

C. Additional Results

In this section, we provide additional result images to further demonstrate the performance and effectiveness of our approach across different models. First, we present additional results for DynamiCrafter [40], an image-to-video model that takes prompts as input (Fig. 7). Furthermore, we provide results for AnimateDiff [14], Lavie [36], and VideoCrafter2 [4].

We observe that our method enhances temporal quality without significantly altering the generated content across these three models. Notably, in Lavie [36], the last frame often deviates significantly from the preceding frames (see rows 3, 6, and 9 in Fig. 10). Our approach effectively mitigates this issue to a large extent.

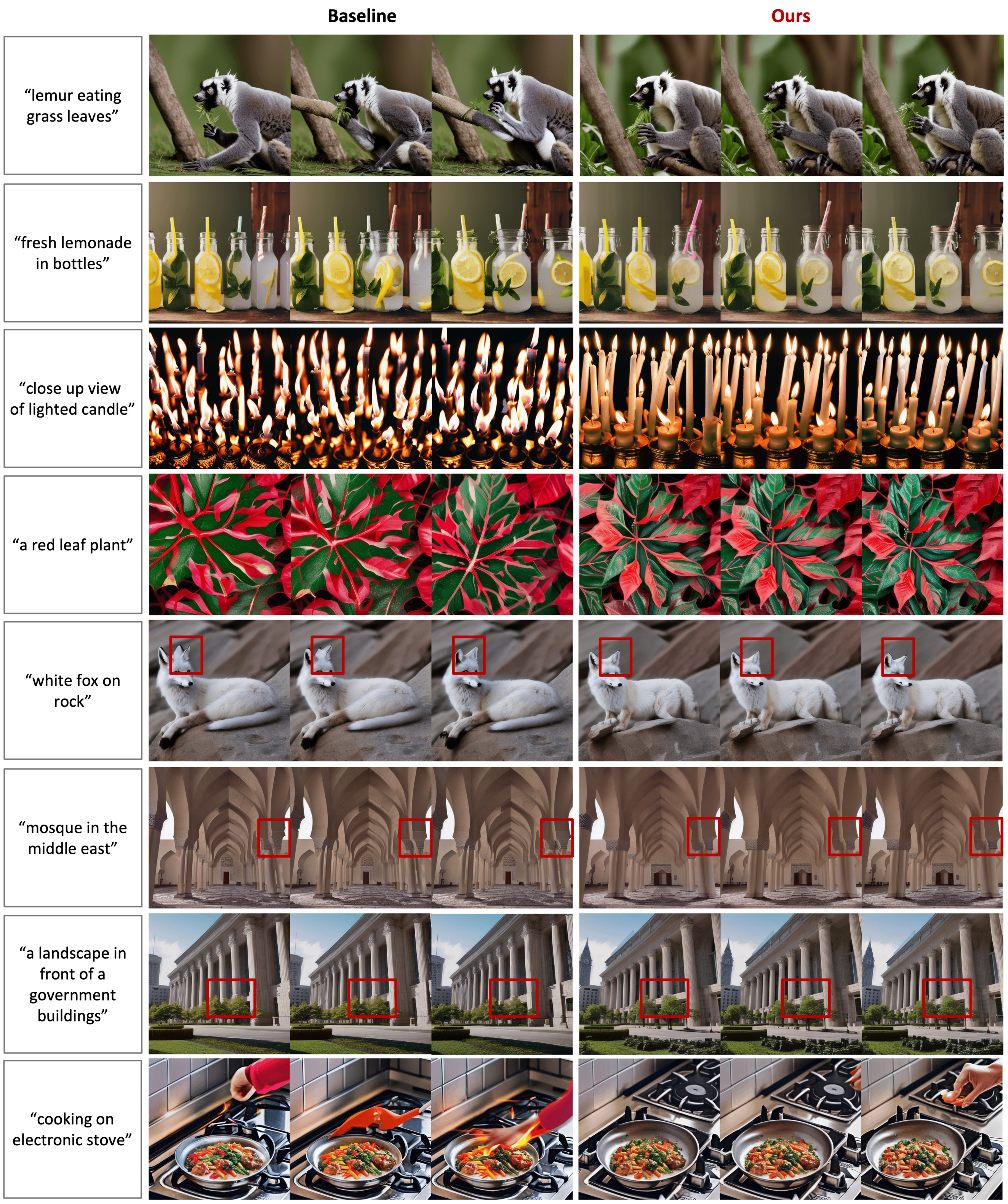
Baseline
Ours
“lemur eating grass leaves”
“fresh lemonade in bottles”
“close up view of lighted candle”
“a red leaf plant”
“white fox on rock”
“mosque in the middle east”
“a landscape in front of a government buildings”
“cooking on electronic stove”

Figure 9. Additional results of AnimateDiff [14].

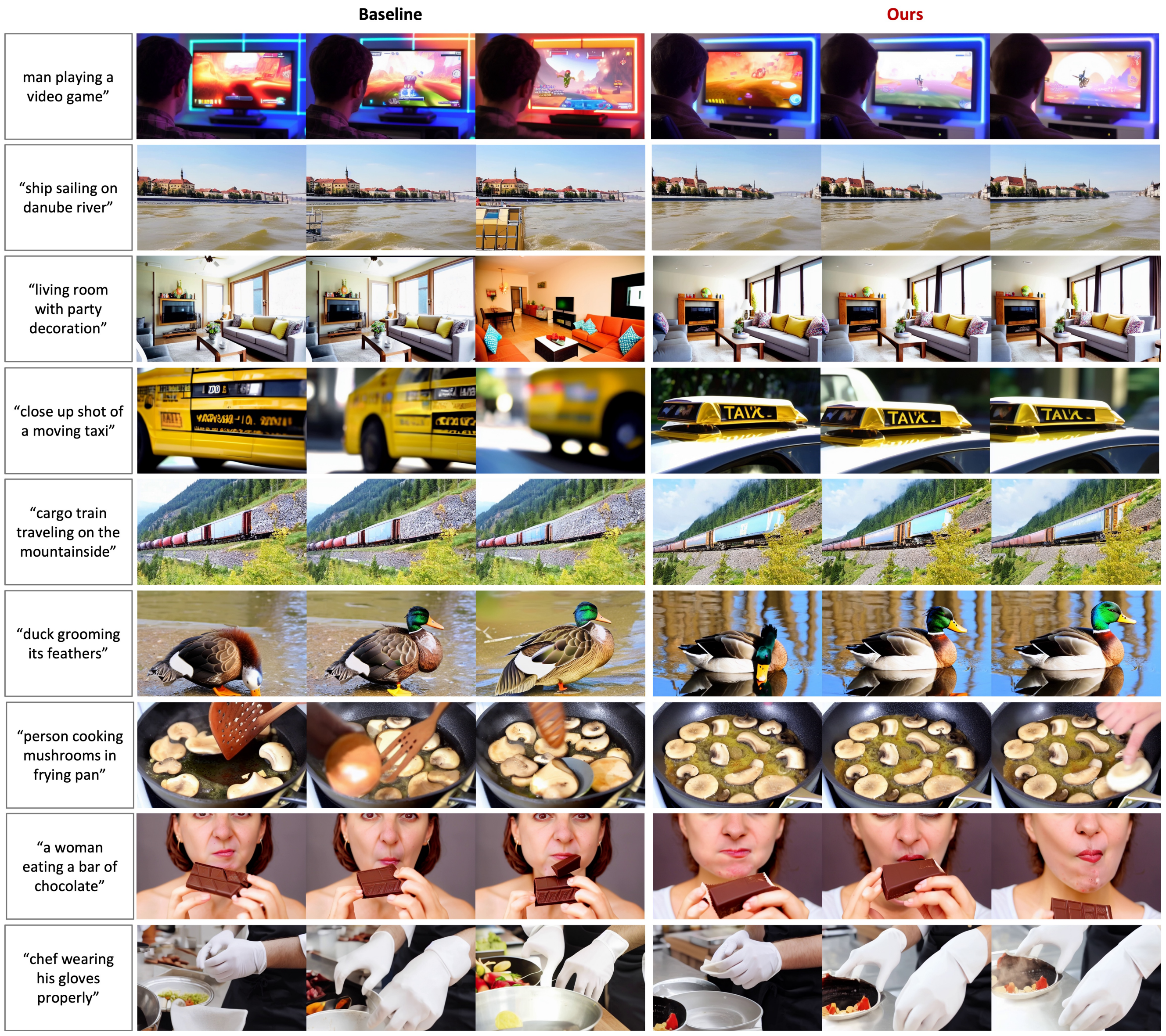
Baseline
Ours
man playing a video game”
“ship sailing on danube river”
“living room with party decoration”
“close up shot of a moving taxi”
“cargo train traveling on the mountainside”
“duck grooming its feathers”
“person cooking mushrooms in frying pan”
“a woman eating a bar of chocolate”
“chef wearing his gloves properly”

Figure 10. Additional results of Lavie [36].

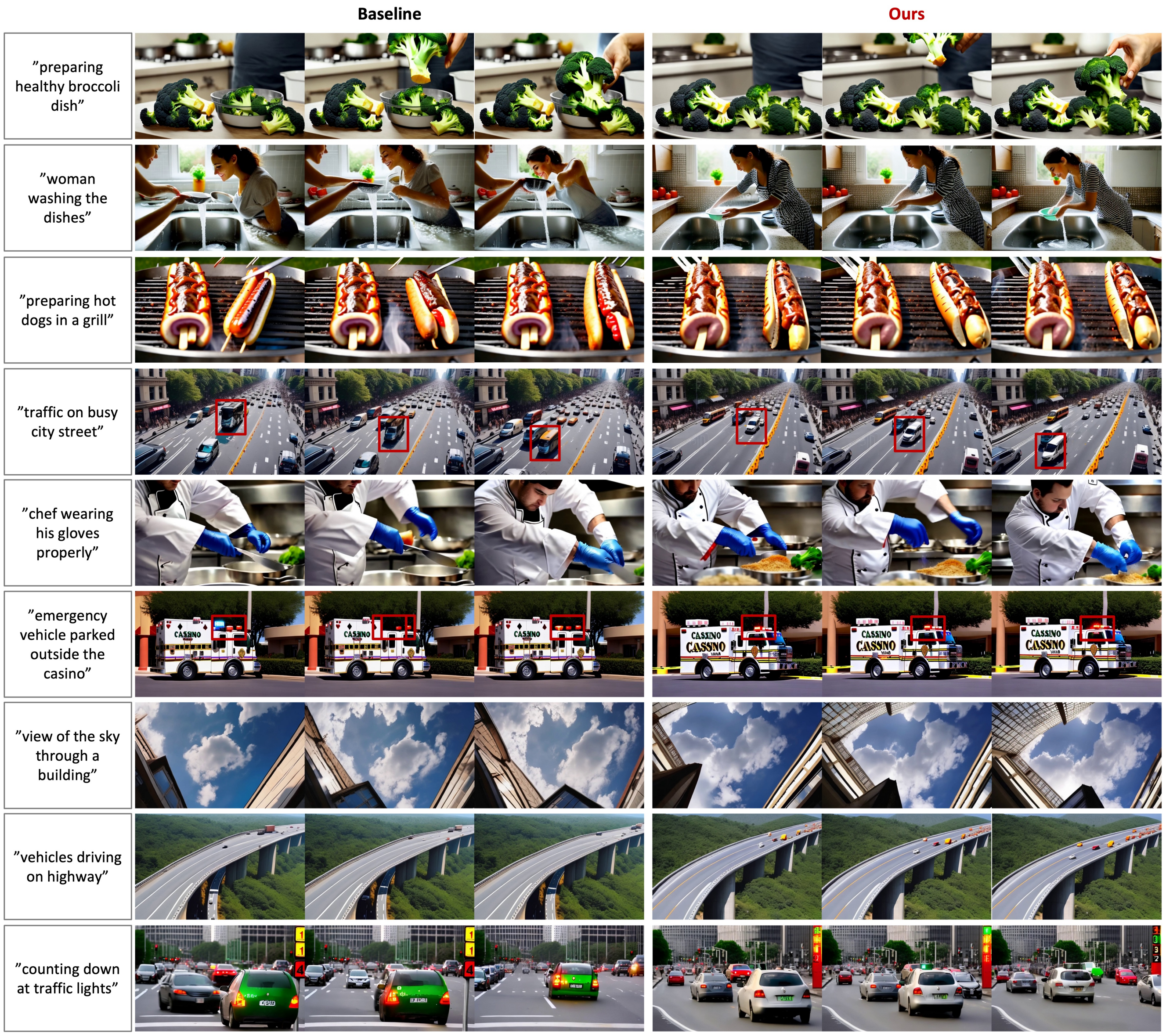
Baseline
Ours
"preparing healthy broccoli dish"
"woman washing the dishes"
"preparing hot dogs in a grill"
"traffic on busy city street"
"chef wearing his gloves properly"
"emergency vehicle parked outside the casino"
"view of the sky through a building"
"vehicles driving on highway"
"counting down at traffic lights"

Figure 11. Additional results of VideoCrafter2 [4].