Encapsulated Composition of Text-to-Image and Text-to-Video Models for High-Quality Video Synthesis

Supplementary Material

1. Overall Evaluation

To emphasis that our method can simultaneously improve imaging quality and temporal consistency, an *overall* score is necessary for consider multiple metrics. As shown in Table 1[1] in the main paper, we calculate an **Overall** score by averaging the normalized values of four scores. Our method achieve the highest score. Note that the normalization range is derived from the VBench LeaderBoard[2], DOVER[3] and Aesthetic Predictor **(AP)** V2.5[4]. The ranges $[\min, \max]$ are as follows:

- MS: $[0.9179, 0.9958]$.
- SC: $[0.8624, 0.9833]$.
- DOVER: $[55.1690, 88.3000]$, where 88.3000 is the score on a real world video dataset, and 55.1690 is the VC2 original video score.
- AP: $[0, 5.5]$: according to the Aesthetic Predictor V2.5 project, $5.5+$ is considered to be a high aesthetic score.

Then, we leverage min-max normalization to compute the scores. The normalized scores are shown in gray in Table 1.

	Temporal (↑)		Imaging (↑)		Overall (↑)
	MS	SC	DOVER	AP	
VC2-ori	0.9829	0.9738	55.1690	4.4975	-
	0.8344	0.9214	0.0000	0.8177	0.6436
Rerender	0.9820	0.9745	76.3923	5.2979	-
	0.8228	0.9272	0.6406	0.9633	0.8385
TokenFlow	0.9696	0.9786	64.4664	4.3898	-
	0.6637	0.9611	0.2808	0.7981	0.6759
BIVDiff	0.9885	0.9801	64.6217	5.0599	-
	0.9063	0.9727	0.2855	0.9182	0.7707
Ours	0.9881	0.9808	73.1997	5.4632	-
	0.9012	0.9793	0.5443	0.9933	**0.8545**

Table 1. Normalized scores reported in Table 1.

2. Ablation Studies

In the main paper, Figure 6 illustrates a case study demonstrating the influence of t_{V} and t_{T2V}. In this section, we conduct comprehensive hyperparameter searching on subset of 20 videos with obvious low imaging quality or temporal inconsistency problem.

[1] Red represents the numbering in the main paper.
[2] https://huggingface.co/spaces/Vchitect/VBench_Leaderboard
[3] https://github.com/VQAssessment/DOVER
[4] https://github.com/discus0434/aesthetic-predictor-v2-5

2.1. Hyperparameter Analysis

There are four hyperparameters mentioned in the *EVS* algorithm (Algorithm 1):

- $t_{\mathrm{I}}(s_{\mathrm{I}} = t_{\mathrm{I}}/T_{\mathrm{I}})$: T2I model noising timestep. Larger t_{I} implies larger noise strength. For the goal of imaging quality enhancement, $t_{\mathrm{I}} = 0.4$ is an optimal choice. Consequently, we have fixed $s_{\mathrm{I}} = 0.4$ in our experiments.
- $t_{\mathrm{V}}(s_{\mathrm{V}} = t_{\mathrm{V}}/T_{\mathrm{V}})$: T2V model noising timestep. A larger t_{V} (larger noise strength) can more effectively eliminate inconsistencies and induce a tendency for frames to converge excessively towards the T2V imaging distribution. We try $s_{\mathrm{V}} \in [0.4, 0.3, 0.2]$.
- t_{T2V}: Timestep of switching to the [T2V] block during T2I denoising steps. A larger value of t_{T2V} (indicating earlier injection of the T2V block during T2I) allows for more timesteps to be available for T2I, resulting in improved imaging quality but increased inconsistency. For $T_{\mathrm{I}} = 30$ setting, we try $t_{\mathrm{T2V}} \in [4, 8, 12]$.
- n_{V}: Number of denoising steps for the [T2V] block. Additional steps yield a more accurate prediction of z_0. However, this is not essential, as the imaging quality is primarily influenced by the T2I denoising processes. We utilize the predicted z_0 as a bridge back to T2I denoising steps. For $T_{\mathrm{V}} = 8$ setting, we try $n_{\mathrm{V}} \in [1, 2]$.

As shown in Table 2, the best hyperparameter combination is $t_{\mathrm{T2V}} = 8, s_{\mathrm{V}} = 0.4, n_{\mathrm{V}} = 2$. The larger s_{V} at 0.4 contributes to improved imaging quality because, despite employing T2V AnimateDiff-Lightning and AnimateLCM for temporal smoothing, these methods also yield high-quality imaging as a complementary benefit.

We can utilize the [T2V] block *multiple* times during the T2I denoising process to explore its potential for further improving temporal consistency. In extreme situations, we can implement this approach at each T2I denoising step. As shown in Figure 1, motion smoothness is determined by the last timesteps of applying the [T2V] block. Multiple applications (e.g., [8, 4], [12, 8, 4], or even at every step [12, 11, ..., 5, 4]) yield minimal improvement over [4] compared to the original video (orange dots), but incur an obvious increase in computational cost (gray bar). The same situation applies for the last timestep with $t_{\mathrm{T2V}} = 8$ (blue dots). The last timestep with $t_{\mathrm{T2V}} = 4$ demonstrates overall better performance compared to $t_{\mathrm{T2V}} = 8$. This confirms our statement that the final temporal consistency of the video is greatly affected by the timing of the last application of the [T2V] block.

		AnimateDiff-Lightning					AnimateLCM				
		Temporal (↑))		Imaging (↑))		Overall (↑))	Temporal (↑))		Imaging (↑))		Overall (↑))
$t_{\text{T2V}}/T_{\text{I}}$	s_{V}, n_{V}	MS	SC	DOVER	AP		MS	SC	DOVER	AP	
4/30	0.4, 1	0.9875	0.9694	82.6544	5.7069	0.9114	0.9818	0.9662	81.9452	5.5904	0.8759
	0.4, 2	0.9892	0.9708	83.0662	5.7434	0.9246	0.9831	0.9672	83.126	5.6014	0.8915
	0.3, 1	0.9840	0.9668	82.6351	5.6788	0.8934	0.9781	0.9653	81.8139	5.5555	0.8596
	0.3, 2	0.9872	0.9692	84.0331	5.6975	0.9200	0.9797	0.9652	81.7247	5.575	0.8647
	0.2, 1	0.9763	0.9635	83.2453	5.6354	0.8645	0.9731	0.9633	81.7129	5.5243	0.8372
	0.2, 2	0.9790	0.9646	83.7943	5.6611	0.8807	0.9744	0.9637	81.7996	5.5421	0.8437
8/30	0.4, 1	0.9853	0.9682	85.0420	5.7229	0.9206	0.9801	0.9658	84.3118	5.5921	0.8875
	0.4, 2	0.9874	0.9692	86.0847	5.7689	**0.9394**	0.9813	0.9667	84.6802	5.6165	**0.8971**
	0.3, 1	0.9804	0.9645	85.5730	5.6845	0.8995	0.9753	0.9642	83.7512	5.557	0.8630
	0.3, 2	0.9841	0.9663	86.0717	5.7050	0.9198	0.977	0.9649	83.5828	5.5788	0.8696
	0.2, 1	0.9732	0.9615	84.6851	5.6385	0.8614	0.9705	0.9619	84.3654	5.5253	0.8460
	0.2, 2	0.9758	0.9617	84.8961	5.6666	0.8730	0.9718	0.9616	83.7408	5.5437	0.8457
12/30	0.4, 1	0.9816	0.9651	86.3023	5.7171	0.9116	0.9753	0.9641	83.9006	5.5737	0.8647
	0.4, 2	0.9846	0.9672	86.2330	5.7501	0.9265	0.9774	0.9648	84.8727	5.5847	0.8807
	0.3, 1	0.9753	0.9617	85.1262	5.6833	0.8739	0.9698	0.9609	83.5959	5.552	0.8371
	0.3, 2	0.9800	0.9647	86.4288	5.7019	0.9059	0.9718	0.9625	84.1262	5.5736	0.8518
	0.2, 1	0.9670	0.9609	83.9691	5.6380	0.8349	0.9644	0.9597	82.9156	5.523	0.8109
	0.2, 2	0.9706	0.9617	85.3942	5.6664	0.8601	0.9662	0.9608	83.9166	5.5404	0.8272

Table 2. The influence of [T2V] block hyperparameters on two T2V base models.

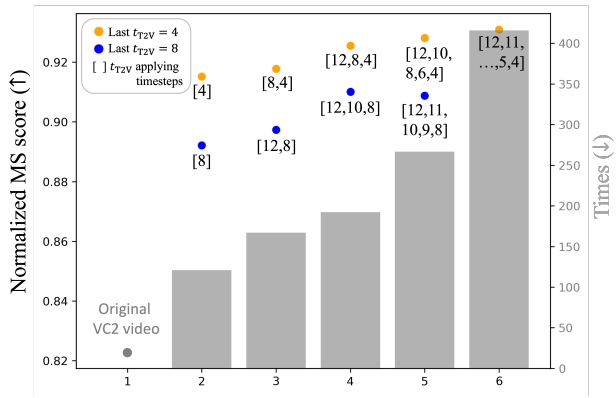

Figure 1. Multiple [T2V] blocks (e.g., [8, 4], [12, 8, 4]) yield minimal MS improvement over [4], but incur an obvious increase in time cost. The last timestep with $t_{\text{T2V}} = 4$ demonstrates overall better performance compared to $t_{\text{T2V}} = 8$.

2.2. Selective Feature Injection Analysis

In this section, we analyze the strategy of *Selective Feature Injection* (SFI), focusing specifically on its ability to leverage the temporal priors of the T2V model. As shown in Figure 2, a basic upsample layer of U-Net in AnimateDiff consists of three parts:

- ResNet: It takes the previous features from the downsample blocks of the U-Net, concatenates them with the current features, and then applies convolution. There are totally 12 layers as shown in the bottom of Figure 3.
- Spatial: Similar to Stable Diffusion, it incorporates Self-Attention (SA), Cross-Attention (CA), and a Feed Forward (FF) Network. There are totally 9 layers, the deepest 3 layers do not have spatial attention modules.
- Temporal: It integrates two SA modules along the temporal dimension, allowing it to aggregate features across different frames at the same spatial coordinates. There are totally 12 layers.

Given a video that has been processed on a frame-by-frame basis using T2I blocks, we employ the DDIM inversion of the T2V model and gather intermediate features from the aforementioned modules. These features can be injected at the input and output of *every* module to establish an upper limit for reconstruction, serving as a starting point for further optical points of partial reconstruction. Each module incorporates a skip-connection structure. Injecting at the input (indicated by the orange arrow in the figure) completely halts the flow of information, thereby preventing the incorporation of temporal priors. Therefore, we choose to inject at the output (as indicated by the blue arrow in the figure). We number all possible injection places with numbers ① to ⑦. [2] injects attention features at ②. They conduct experiments and discover that Self-Attention (SA) in the deeper layers of U-Net effectively captures the structural information of an image. PnP [3] additionally injects features at ① at deeper U-Net layers for better preservation of structural information. For video diffusion models, reconstruction poses significant challenges, largely due to the limitations in training data and the inherent denoising capabilities of the model. Additionally, incorporating extra temporal modules creates more potential places for us to inject features.

Figure 3 illustrates the reconstruction PSNR for different layer IDs (representing injection from layer 0 to each specified layer) at the seven output locations above. Directly

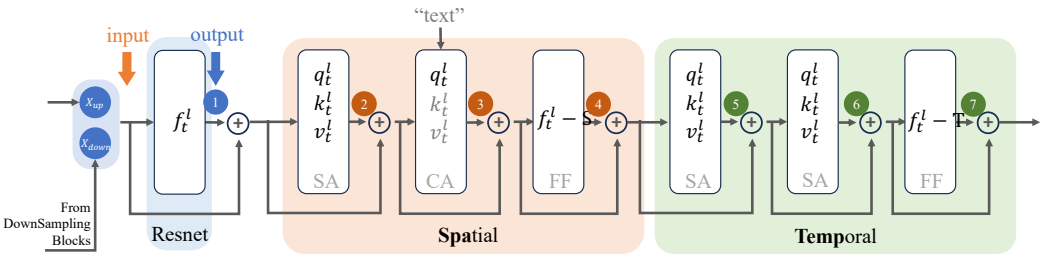

Figure 2. Illustration of a basic upsample block in the T2V model (a.k.a, layer), which consists many skip-connections. We number the places where we can inject the features. One can inject feature at input or output, while injecting features at input will totally prevent previous features be transport to the current position.

injecting at the shallowest ResNet layer reaches the upper limit of reconstruction. However, injecting at ResNet layers results in a steady increase at the last two layers. This makes it challenging to identify an optimal balance. Injecting at either the spatial or temporal layer yields a smooth increment. The spatial layer achieves a higher PSNR. Therefore, we have chosen the spatial layer as the preferred injection location (②). Starting from injecting into all spatial layers, we identify optimal points to balance the preservation of imaging quality with the introduction of temporal prior, as shown in Figure 8 in the main paper. For visualization results on *stylized* video from VBench, see Figure 4.

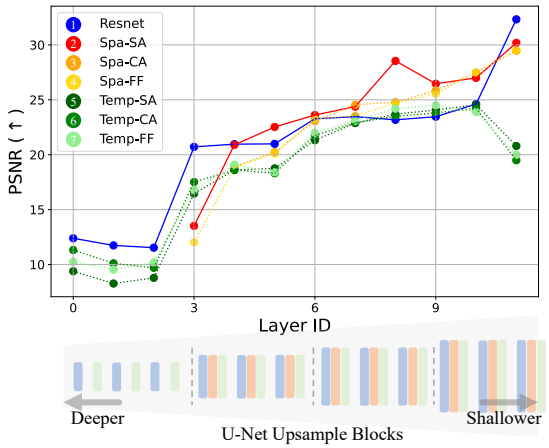

Figure 3. Seletive Feature Injection (SFI) at different module output (① to ⑦) and layers (totally 12 layers).

3. More Results

3.1. More Baselines

Figure 5 presents a visualization of four additional baselines. FRESCO improves the use of optical flow for feature propagation compared to Rerender-A-Video. However, the performance is ultimately constrained by inconsistencies present in the original video. TokenFlow combined with PnP [3] editing leverages feature matching derived from the inverted source video with greater accuracy. However, this approach results in reduced editability when compared to SDEdit. As illustrated on the right side of Figure 5, the color inconsistency of the car persists and cannot be effectively resolved. RAVE [1], proposes the randomized noise shuffling method, suffering the issue of flickering fine details. AnyV2V, even with an increased number of inversion steps (ranging from 100 to 500), does not demonstrate improved performance. The I2V model struggles to fully propagate the edits made to the first frame into the subsequent frames because it must integrate features such as PnP to maintain the original video's motion. This necessity inadvertently introduces original imaging content. Figure 6 presents another two case studies.

3.2. Combined with Developing T2V Base Models.

With increasingly advanced architectural designs and the inclusion of more training videos, both motion quality and imaging quality have demonstrated continuous improvement. However, when video datasets predominantly composed of real-world scenes are used, training T2V models inevitably leads to a lower capacity for image generation compared to T2I models. Even though models such as CogVideoX and HunyuanVideo achieve high aesthetic scores for common life scenes, they sacrifice the understanding of text prompts related to imaginary scenarios (e.g.astronaut riding horse) or complex visual descriptions (e.g.style of VanGogh). As demonstrated in VBench (Up to January 2025), VideoCrafter-2.0 still outperforms all other T2V models, including closed-source ones, in terms of Appearance Style (image-text CLIP similarity). There still remains potential for enhancing imaging quality across these diverse scenarios as T2I models continue to be improved. We provide the result of CogVideoX-5B in combination with our EVS in Figure. 3. For HunyuanVideo/closed-source, their high-quality realistic video generation can further be integrated with our EVS editing framework to address current limitations in uncommon style understanding.

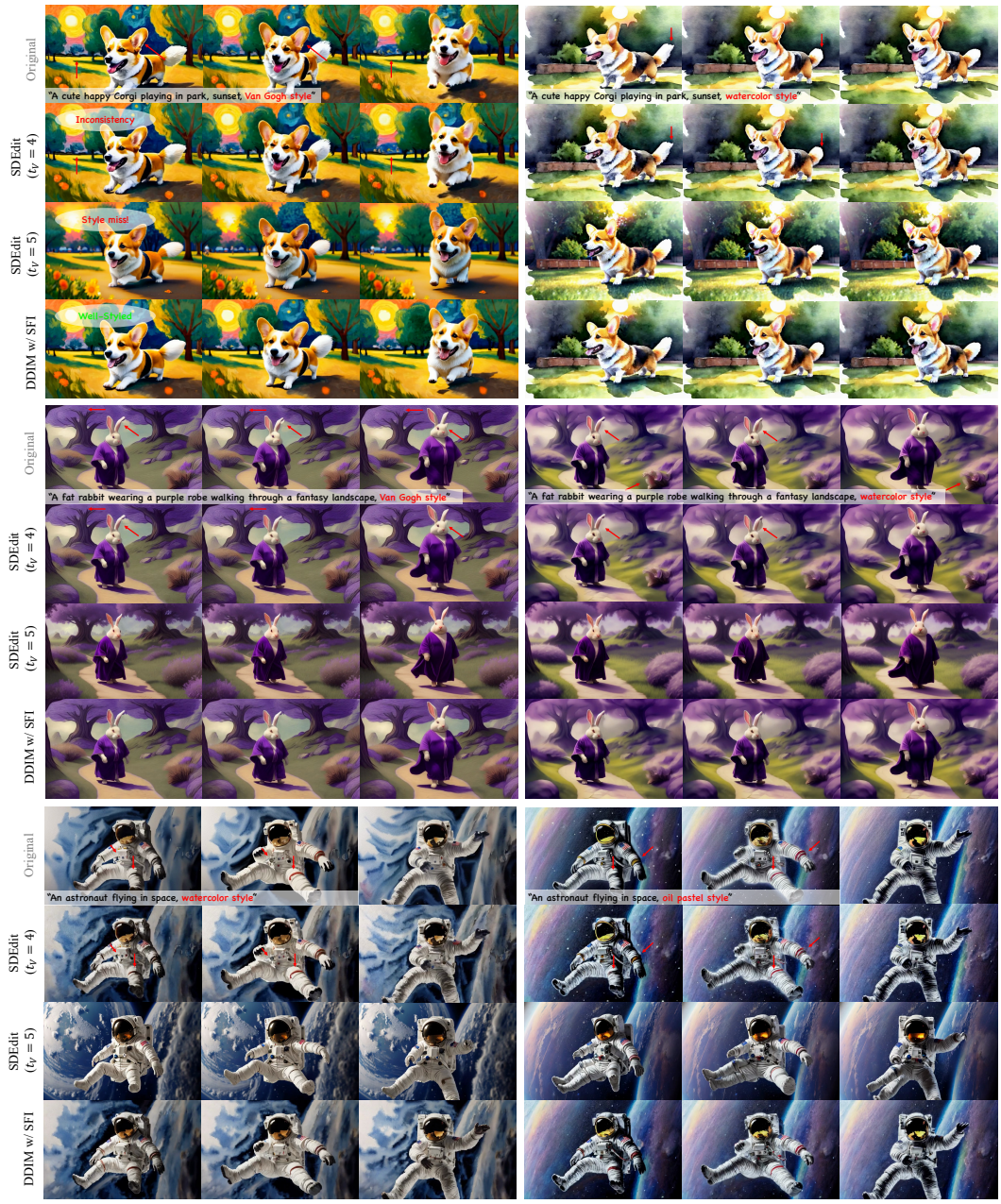

Figure 4. DDIM with Selective Feature Injection (SFI) vs. SDEdit with varying noising strengths. For stylized videos that lie outside the imaging domain of the T2V model, a balanced noising point is absent. Notably, between $t_V = 4$ to $t_V = 5$, there is a sudden loss of stylization. In contrast, our DDIM with SFI effectively preserves the stylization while mitigating inconsistencies.

In the next section, we will demonstrate editing ability of EVS.

	MS (↑)	SC (↑)	DOVER (↑)	AP (↑)	DD (↑)
CogVideoX-5B	0.9749	0.9548	54.26	4.63	0.5667
CogVideoX-5B+RV	0.9818	0.9659	56.53	5.29	0.5652

Table 3. CogVideoX generated video from VBench combined with RealisticVisionV60B1 (RV) under our EVS framework.

3.3. Video Editing

Our approach to the generative video quality enhancement task can also be applied to real-world video editing, or API generated video post-processing tasks. Figures 7 and 8 present the results of two classic case study datasets. The baseline results, with the exception of AnyV2V, were sourced from the FRESCO webpage. In Figure 7, our method demonstrates superior consistency in detail, partic-

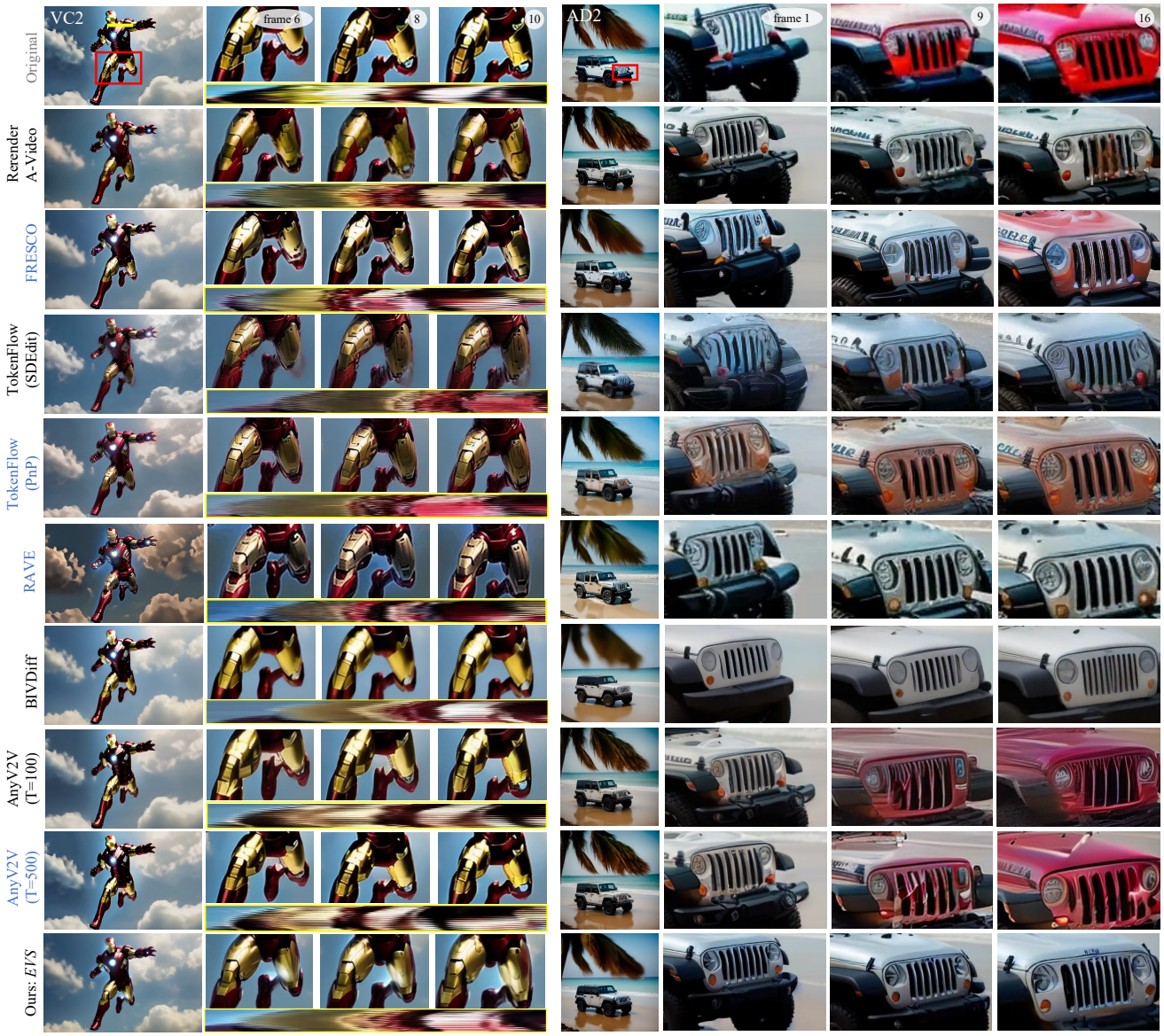

Figure 5. Blue indicates newly added visualization compared with Figure 5 in the main paper.

ularly highlighted in the zoomed-in red box where the nail is located. In Figure 8, the baselines select frames from the original video at intervals of 5. As a result, the motion between frames is discontinuous and falls outside the motion domain of the T2V model. Nonetheless, directly applying T2V smoothing demonstrates superior motion smoothness compared to most baseline methods, particularly in the background (as highlighted by the zoomed-in yellow line, our video exhibits noticeably fewer flickers).

References

[1] Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6507–6516, 2024. 3

[2] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7817–7826, 2024. 2

[3] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 2, 3

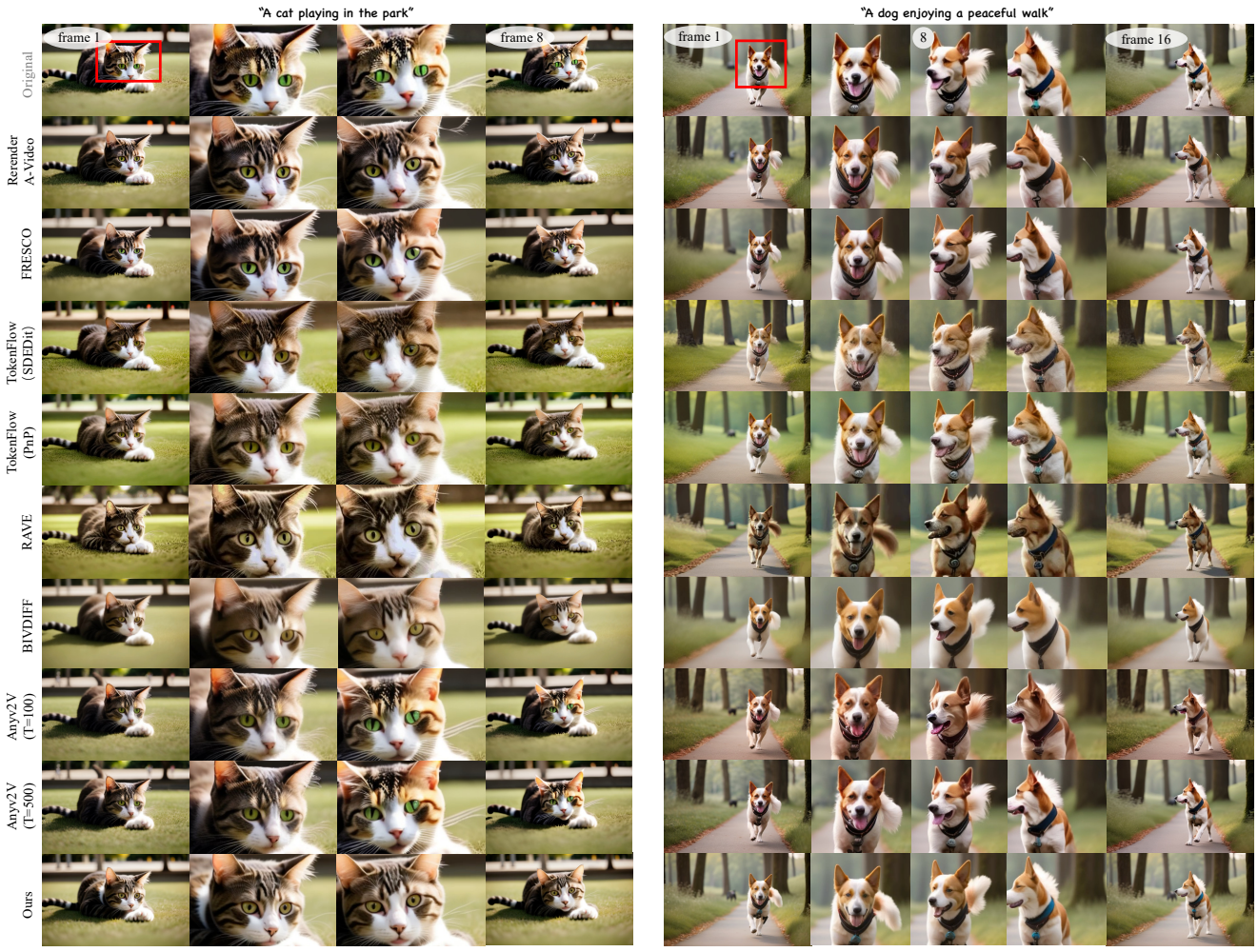

Figure 6. Additional case studies on VBench, VideoCrafter-2, Subjective Consistency subset of videos, compared with baseline methods.

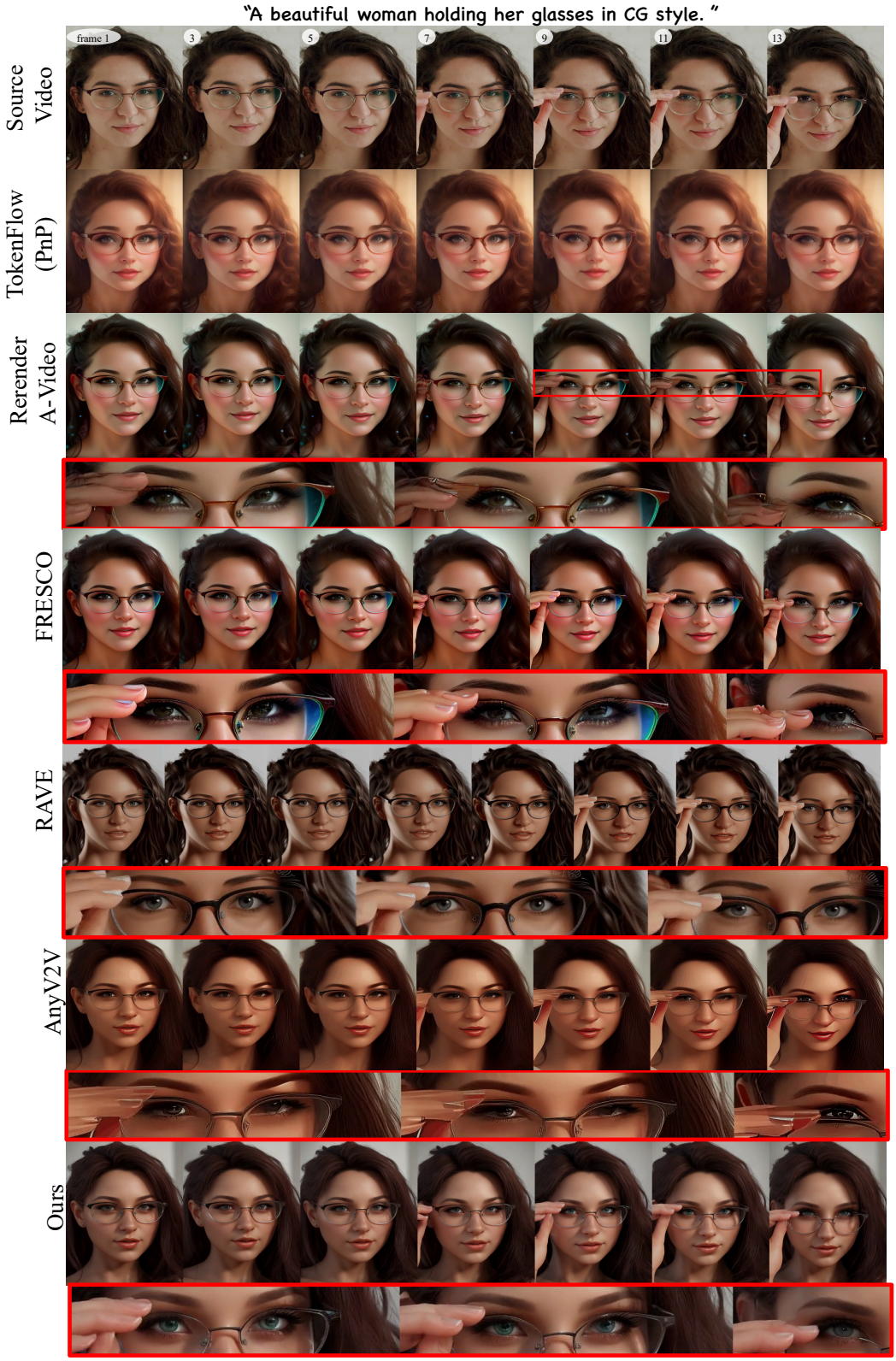

Figure 7. Real-world video editing. Baseline results are downloaded from FRESCO webpage. The zoomed-in red box highlights the superior consistency of our methods. Notably, TokenFlow fails to capture the finger in the source video.

Figure 8. Real-world video editing. Baseline results are downloaded from FRESCO webpage. Control-Video, T2V-Zero and TokenFlow show obvious artifacts and blurs. The yellow lines stacked across frames for the other baselines indicate that our method achieves superior motion smoothness.