

# FiVE-Bench : A Fine-grained Video Editing Benchmark for Evaluating Emerging Diffusion and Rectified Flow Models – Supplementary Materials

Minghan Li<sup>1\*</sup>, Chenxi Xie<sup>2\*</sup>, Yichen Wu<sup>1,3</sup>, Lei Zhang<sup>2</sup> and Mengyu Wang<sup>1†</sup>

<sup>1</sup>Harvard University

<sup>2</sup>Hong Kong Polytechnic University, <sup>3</sup>City University of Hong Kong

{mili4, mengyu.wang}@meee.harvard.edu

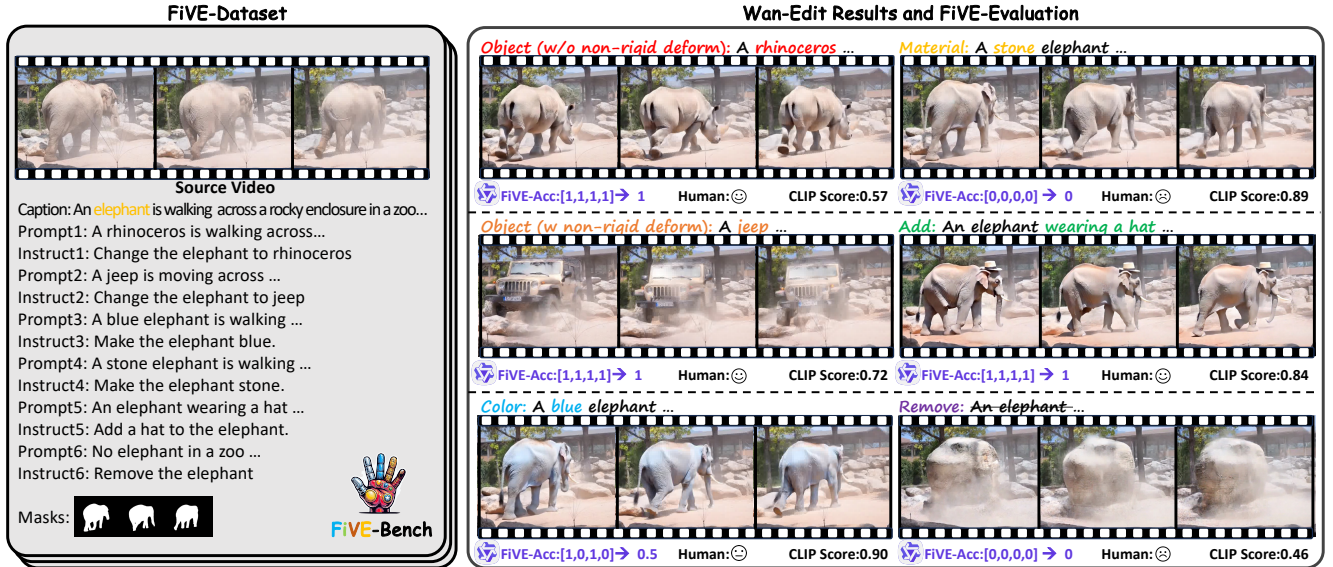


Figure 1. Overview of the proposed **FiVE-Bench**, with editing results from **Wan-Edit** and corresponding **FiVE-Acc** scores.

In this supplementary file, we provide the following materials:

- Additional details on baseline methods
- Expanded implementation details
- Further explanation of the FiVE benchmark
- Human preference validation
- GPU memory usage and speed comparison
- Additional quantitative results and analysis
- Additional qualitative results and analysis

## A. Baseline Methods

- TokenFlow [2] is a training-free framework for consistent video editing that leverages diffusion features by enforcing cross-frame semantic token alignment in latent space to preserve spatiotemporal coherence. By propagating consistent appearance and motion patterns through op-

timized token interactions in a pre-trained text-to-image model, it achieves temporally stable edits without requiring additional fine-tuning or annotated data.

- DMT [9] is a zero-shot framework for text-driven human motion transfer that leverages spatiotemporal diffusion features to align source motion patterns with target textual descriptions in a unified latent space. By integrating cross-modal attention mechanisms and temporal coherence constraints within a pre-trained diffusion model, it enables realistic motion synthesis without requiring task-specific training or paired data, ensuring both semantic fidelity and dynamic consistency.
- VidToME [5] is a zero-shot video editing framework that enhances spatiotemporal consistency by adaptively merging redundant tokens across frames within a pre-trained diffusion model. This token-efficient strategy preserves critical motion and appearance features while reducing computational overhead, enabling coherent video edits

\*Equal contribution, † Corresponding author.

Table 1. Comparison of different video editing methods for DM and RF models under default settings.

Methods	Publication	Inv. Type	Attention Injection	Base T2I/V Model	Inv.-free	Resolution	Timesteps Inv.+Edit	Conditions
DMs	TokenFlow	DDIM	✓	SD2.1	✗	(512, 512)	500 + 50	✗
	DMT	DDIM	✗	ZeroScope	✗	(576, 320)	1000 + 50	Optimization
	VidtoMe	PnP	✓	SD1.5	✗	(512, 512)	50 + 50	✗
	AnyV2V	PnP	✓	I2VGen-XL	✗	(512, 512)	500 + 50	InstructPix2Pix
	VideoGrain	DDIM	✓	SD1.5	✗	(512, 512)	50 + 50	Depth + Mask
RFs	Pyramid-Edit	Ours	FlowEdit	Pyramid-Flow	✓	(640, 384)	40	✗
	Wan-Edit	Ours	FlowEdit	Wan2.1	✓	(832, 480)	50	✗

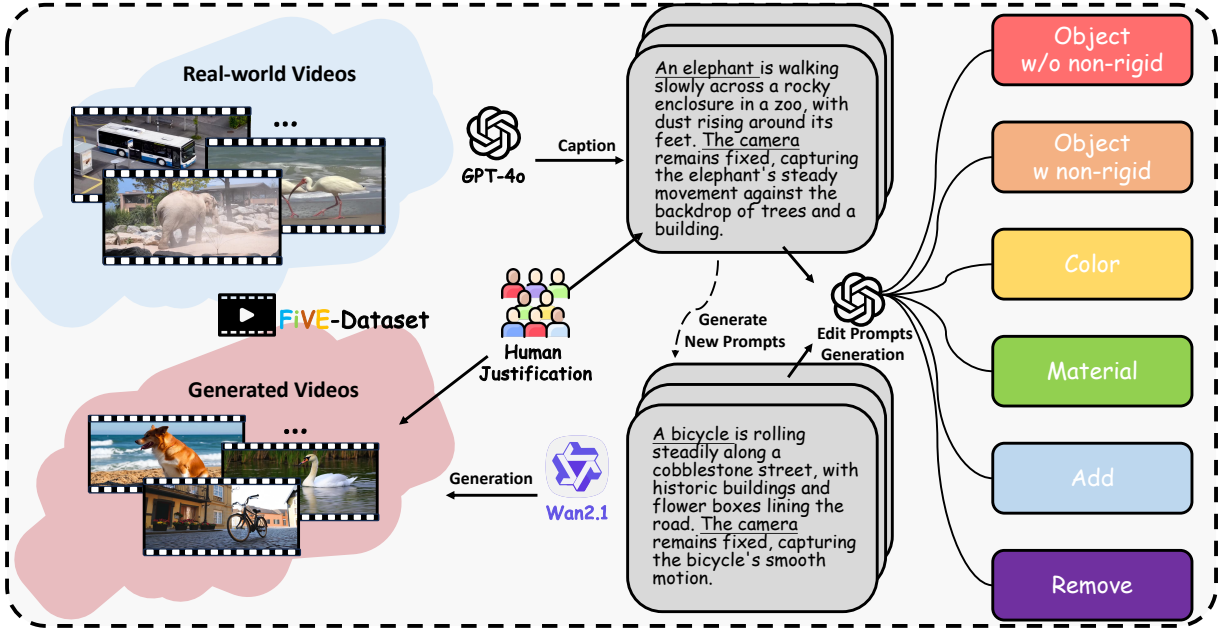


Figure 2. FiVE-Benchmark construction pipeline, involving the collection of real-world videos, the generation of captions for these videos, the creation of synthetic video-caption pairs, and the generation of editing prompts.

without task-specific training or temporal-aware fine-tuning.

- AnyV2V [4] is a tuning-free framework designed for universal video editing tasks, leveraging spatiotemporally consistent diffusion features through cross-frame latent propagation to maintain coherence across diverse editing operations. By dynamically aligning semantic and motion patterns in pre-trained diffusion models without task-specific tuning, it enables flexible video-to-video transformations while preserving temporal stability and visual fidelity. For prompt-based editing, it uses Instruct-Pix2Pix [1] to edit the first frame first.
- VideoGrain [8] is a video editing framework that enables multi-grained control through hierarchical space-time attention modulation, dynamically adjusting spatial and temporal feature interactions in diffusion mod-

els to achieve precise edits across varying granularities. By decomposing and recombining cross-frame attention patterns at different resolution scales, it maintains temporal coherence and visual fidelity while supporting diverse editing tasks without requiring architectural modifications or task-specific fine-tuning.

## B. Implementation Details

All experiments were conducted using the official GitHub repository and environment, with default settings. For training-free methods, the editing results are highly dependent on hyperparameters, such as in PNP [7]. To minimize the impact of hyperparameters, we randomly selected six videos from our benchmark and performed a search within an appropriate parameter range to find the best hyperparameters. These were then fixed for all subsequent data in the

benchmark. All experiments were run on a single H100 GPU. Table 1 lists the parameter settings for the compared methods and our proposed approach.

For VLM-based FiVE-Acc evaluation, QWen2.5-VL-7B is selected as the evaluation model. We sample one frame every 8 frames from the edited video, selecting a total of 5 evenly spaced frames from a 40-frame video, which are then fed into the vision encoder of QWen2.5-VL. The text input consists of Yes/No questions or multiple-choice questions, as illustrated in Fig. 2 of the main paper. Considering the varying number of videos across different editing types, we compute the FiVE-Acc metric separately for each type. The final FiVE benchmark score is obtained by averaging the scores across all six editing types, as presented in Table 3 of the main paper. The evaluation codebase is available at: <https://sites.google.com/view/five-benchmark>.

### C. FiVE Benchmark

The construction of the FiVE-Benchmark involves the collection of real-world videos, the generation of captions for these videos, the creation of synthetic video-caption pairs, and the generation of editing prompts. The overall pipeline is illustrated in Fig. 2.

#### C.1. Video-description Pair Construction.

We begin by selecting real-world videos from the DAVIS dataset [6] that are well-suited for fine-grained video editing. For each chosen video, we use GPT-4o [3] to generate detailed annotations every 8th frame, capturing key elements such as subject actions, background details, and camera movements. Next, we create new annotations in the style of real video descriptions, which are then used to guide a text-to-video model in generating new videos. The full process and examples of generated caption-video pairs are shown in Fig. 3. In this process, human justification is involved in assessing the quality of both the videos and their descriptions to ensure the generation of high-quality video-description pairs.

#### C.2. Editing Prompt Generation

For the constructed video-caption pairs, we design specialized prompts to generate target editing instructions for six editing types. GPT-4o is employed to create new video captions by modifying the original captions based on the target object, serving as the target prompts for the editing process. Fig. 4 provides an example of the prompt and its corresponding output for generating an editing type instruction.

#### C.3. FiVE-Acc Question Generation

The FiVE benchmark provides both the source object (the object in the original video) and the target object (the object after editing). Based on this information, we use GPT-4o to

generate Yes/No questions and Multiple-choice questions, as illustrated in Fig. 5. To improve question quality, the user prompts for GPT-4o are customized according to the editing type, and all generated questions are manually reviewed for accuracy. These questions are then used to assess the success rate of video editing methods, serving as the foundation for the FiVE-Acc metric.

### D. Human Preference Validation

To ensure the reliability and perceptual alignment of our evaluation protocol, we conduct comprehensive human preference validation on three aspects: the proposed FiVE-Acc metric, the compared editing methods, and the quality of segmentation masks.

**Human validation of the FiVE-Acc metric.** In Fig. 6 (left), we conducted a human study on 16 randomly sampled videos with 64 prompts. As shown in Table ?? of the main paper, the overall FiVE-Acc score derived from human judgments (47.34%) closely aligns with the automatic evaluation by Qwen-2.5-VL (46.97%), showing only a marginal difference of +0.37%. This strong agreement suggests that the proposed FiVE-Acc metric is well-aligned with human perception.

**Human preference across editing methods.** In Fig. 6 (central), we randomly sample 11 videos, resulting in 45 source–target prompt pairs. The seven methods included in the study correspond to those listed in Table 2 (methods a–g) of the main paper. To reduce bias, the order of methods is randomized. Annotators are asked to select their top-2 preferred results per prompt, and the aggregated votes are used to compute preference statistics in Fig. 7. The left pie chart shows the percentage of times each method was selected as the Top-1 only preference by annotators. Method g (our Wan-Edit) clearly dominates with 66.2% of Top-1 selections, indicating strong perceptual preference. The right pie chart presents the combined percentage of times each method was selected as either Top-1 or Top-2, providing a broader measure of perceived quality. Again, method g (our Wan-Edit) leads with 41.9%, followed by method e (VideoGrain: 20.6%) and method b (DMT: 15.1%). These results highlight the overall superiority of our Wan-Edit in human perception, while also reflecting relative strengths of other methods in terms of broader acceptance.

**Human validation of mask quality.** The segmentation masks in FiVE benchmark are initially generated by the SAM model to provide object-level supervision. To ensure accuracy and alignment with the intended target regions, each mask is manually reviewed and corrected if necessary by human annotators (see Fig. 8). This process guarantees high-quality annotations that support reliable evaluation.

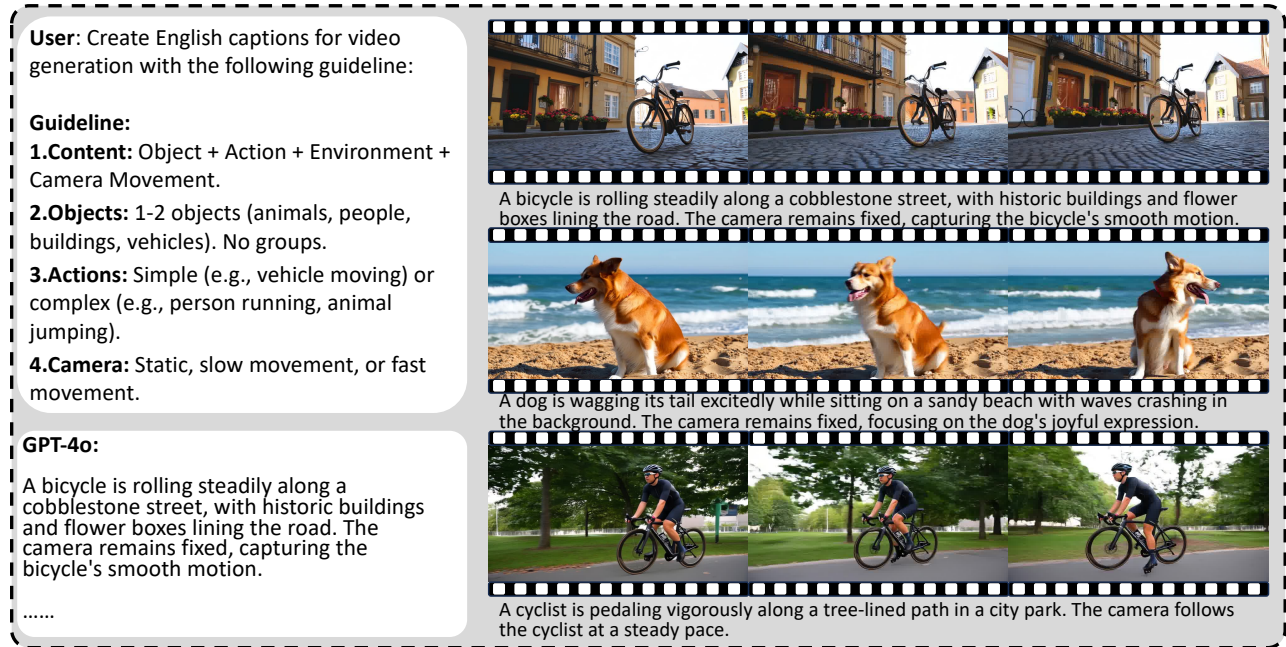


Figure 3. Example of generated caption-video pairs.

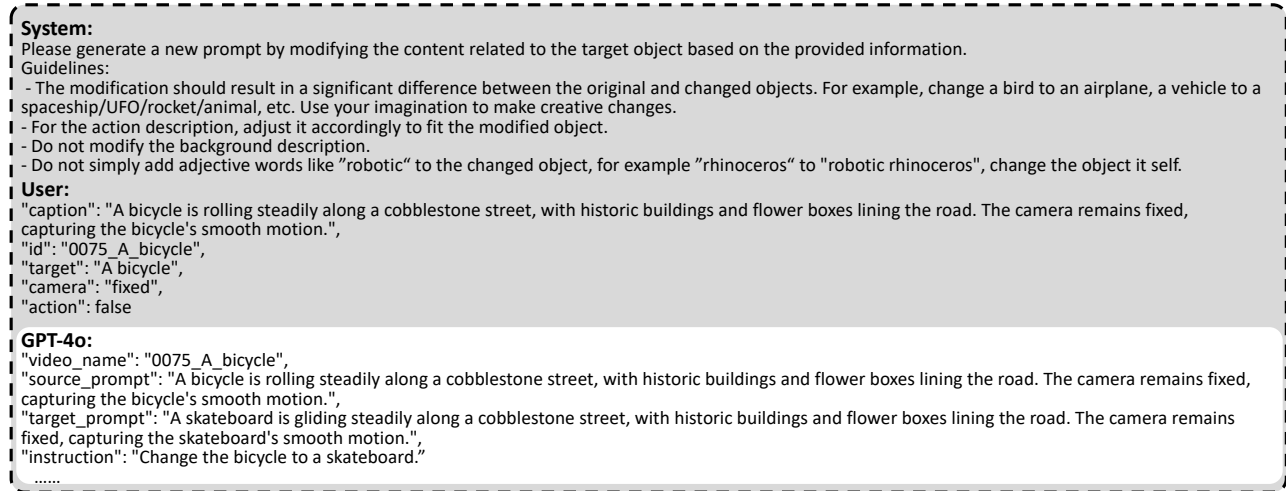


Figure 4. Example of editing prompt generation.

## E. GPU Memory and Speed

As shown in Fig. 9, TokenFlow, VidToMe, AnyV2V, and our proposed Wan-Edit are all positioned in the lower-left corner, indicating a well-balanced trade-off between editing time and peak memory usage. However, Wan-Edit significantly outperforms the other three in terms of editing quality, further demonstrating its effectiveness. Compared to the highly competitive VideoGrain, Pyramid-Edit and Wan-Edit drastically reduces editing time and memory consumption, proving their efficiency.

The speed differences among these methods stem from

their architectural choices and computational requirements. Pyramid-Edit is the fastest, benefiting from its multi-resolution design and the high spatiotemporal compression rate of VideoVAE ( $8 \times 8 \times 8$ ), which significantly reduces the processing burden. However, this aggressive compression can lead to background collapse, particularly when the background exhibits fast motion. Wan-Edit, on the other hand, adopts a more moderate compression rate ( $4 \times 8 \times 8$ ), which balances efficiency and quality, ensuring better background preservation while maintaining competitive speed. In contrast, VideoGrain is significantly slower due to its de-

**System prompt:**

Given the source and target prompts, along with the source and target objects, generate a question about the edited object that reflects its transformation from the source to the target.

**User example: (customized for each editing type)**

Source prompt: 'A black swan swimming in the river.'

Target prompt: 'A flamingo swimming in the river.'

Source object: 'A black swan'

Target object: 'A flamingo'

Yes/No Questions: 'Is that a black swan in the river?' \n 'Is that a flamingo in the river?'\n\n

Multi-choice Question: 'What is the object in the river? Options: A) Black swan B) Flamingo'

**GPT-4o:**

Source prompt: '{source prompt}'\n

Target prompt: '{target prompt}'\n

Source object: '{source object}'\n

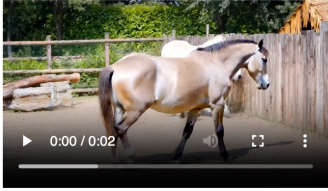
Target object: '{target object}'\n

Yes/No Questions:

Multi-choice Question:

Figure 5. Example of Yes/No and Multi-choice question generation for FiVE-Acc evaluation.

0036\_camel



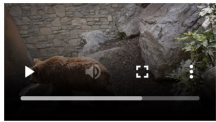
**Q1: Is the animal a camel in the video?**  
☐ Yes  
☐ No

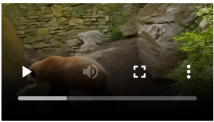
**Q2: Is the animal a horse in the video?**  
☐ Yes  
☐ No

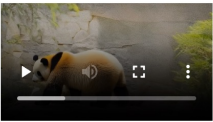
**Q3: What type of animal is walking along the fence in the video?**  
☐ a) A camel  
☐ b) A horse  
☐ c) Neither


**Prompts:**

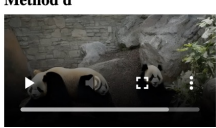
- Source prompt: a large brown bear
- Target prompt: a large panda

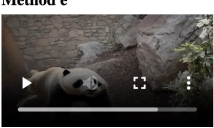
Input  


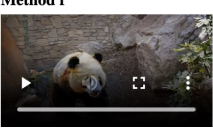
Method a  


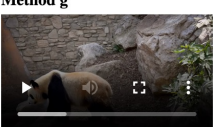
Method b  


Method c  


Method d  


Method e  


Method f  


Method g  


Before selection, please watch the videos and make your best judgement based on **Fidelity**, **Editing**, and **Quality**.

**Top 1: Which method is the best?**      **Top 2: Which method is the second best?**

☐ a ☐ b ☐ c ☐ d ☐ e ☐ f ☐ g      ☐ a ☐ b ☐ c ☐ d ☐ e ☐ f ☐ g

Figure 6. Human evaluation example using [Netlify](#). *Left*: An example illustrating human verification of the FiVE-Acc metric, conducted on Wan-Edit results. *Right*: A human preference study where annotators are asked to select the top-2 preferred results.

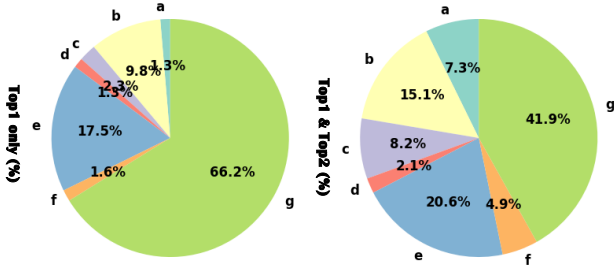


Figure 7. Human preference statistics for the evaluated editing methods (a–g) listed in Table 2 of the main paper.

ditional processing steps, which involve extracting object masks and depth maps to guide the editing process, introduce substantial computational overhead. This makes VideoGrain less suitable for real-time or high-speed applications despite its strong editing accuracy.

## F. More Quantitative Results and Analysis

In this section, we present additional experimental results and provide a comprehensive analysis of the performance of all baseline methods on our proposed FiVE benchmark. For clarity, we define six editing types, referred to as Edit 1–6: rigid transformation (e.g., car to bus), non-rigid transformation (e.g., car to elephant), color change (e.g., black to red), attribute change (e.g., car to a wooden texture), ob-

pendence on segmentation and depth models. These ad-

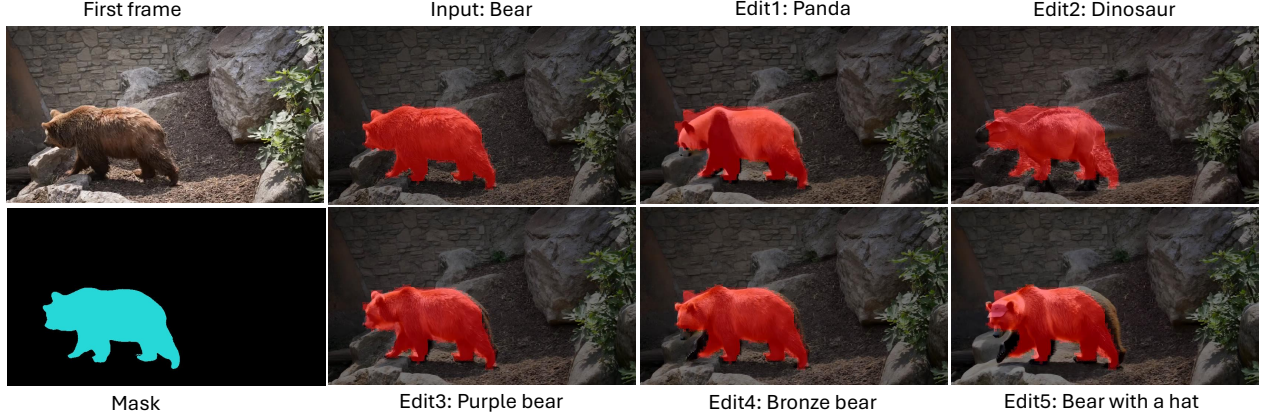


Figure 8. Human validation of SAM2-predicted mask quality on all six edit types.

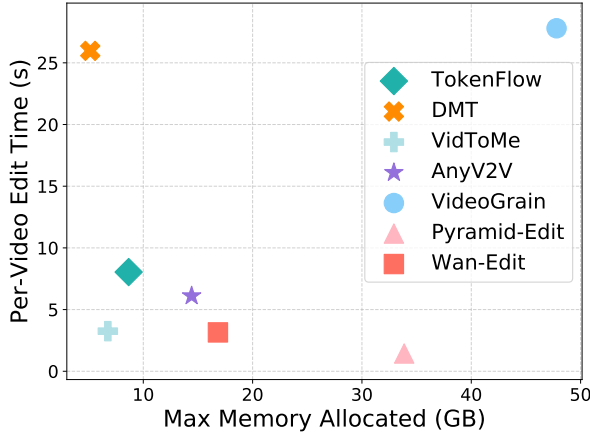


Figure 9. Comparison of editing efficiency, including GPU memory usage and per-frame running time. All test on a single NVIDIA H100.

object addition, and object removal. We conduct a comparative analysis of various video editing methods across these categories using multiple metrics. Additionally, we evaluate diffusion- and flow-based approaches on the proposed FiVE benchmark and FiVE-Acc metric.

Tables 2 and 4 present the results for **Edit1**: Object replacement without non-rigid transformations (e.g., replacing a car with a bus). Our proposed Wan-Edit achieves background preservation and text alignment comparable to VideoGrain, while exhibiting superior motion fidelity. Since Wan-Edit better retains the original video background, its IQA scores remain consistent with those of the source video. Regarding the FiVE-Acc metrics, which assess the accuracy of successful edits, the most competitive method is DMT, which optimizes editing based on the input text. The training-free Wan-Edit and VideoGrain achieve similar results, slightly trailing DMT but significantly outperforming other methods. Overall, DMT offers the best

text-vision alignment but requires optimization, whereas VideoGrain and Wan-Edit strike a strong balance across various fine-grained video editing metrics. Notably, Wan-Edit stands out for its superior efficiency, delivering faster and more stable results compared to VideoGrain.

Tables 3 and 5 present the results of all compared methods on **Edit2**: Object Replacement with Non-Rigid Transformations, revealing similar conclusions to **Edit1**: Object Replacement with Rigid Transformations. However, **Edit2** is more challenging than **Edit1** due to the complexity of non-rigid transformations. This increased difficulty results in a noticeable drop in text-vision alignment, motion fidelity scores, and the editing success rate (FiVE-Acc) metrics compared to **Edit1**. In terms of the editing success rate (FiVE-Acc), DMT and Wan-Edit achieve 67.86% and 52.02%, respectively, on **Edit1**, with DMT outperforming Wan-Edit by approximately 15%. However, on **Edit2**, DMT drops significantly to 53.72%, while Wan-Edit remains stable. This indicates Wan-Edit’s robustness in handling non-rigid transformations, maintaining consistent performance even in more challenging editing scenarios.

Similarly, consistent trends are observed across other editing types, **Edit3** (color changes) and **Edit4** (object material changes) as shown in Tables 6 - 9, further reinforcing our conclusions. Regarding the FiVE-Acc metric, color changes (**Edit3**) achieve the highest editing success rate among all types, with VideoGrain reaching 86% and Wan-Edit at 63%. In contrast, object material changes (**Edit4**) are significantly more challenging, with AnyV2V achieving the highest success rate at 43%, followed by Pyramid-Edit at 36%. This difficulty arises because object material changes often require modifying mid- and low-frequency noise, making training-free methods highly sensitive to parameters, which leads to lower success rates.

Tables 10 - 13 compare the results of **Edit5** (object addition) and **Edit6** (object removal). For object addition (**Edit5**), VideoGrain and Wan-Edit achieve the high-

est scores in background preservation and motion fidelity, while TokenFlow performs best in text-vision alignment and IQA. In terms of FiVE-Acc, the RF-based methods Pyramid-Edit and Wan-Edit achieve success rates of 83% and 72%, respectively, whereas the highest-performing diffusion-based method, DMT, reaches only 61%, highlighting the advantage of RF-based approaches in this task. For [Edit6](#) (object removal), nearly all methods perform the worst among all editing types, with FiVE-Acc scores dropping below 20%, indicating that object removal is one of the most challenging fine-grained editing tasks. This difficulty arises because removing an object requires precisely inpainting the occluded background while maintaining temporal coherence, which is particularly challenging for existing editing models. The visualizations in Fig. 4 of the main paper further confirm these findings.

In conclusion, our analysis ranks the difficulty of fine-grained video editing tasks, with color changes ([Edit3](#)) being the easiest and object removal ([Edit6](#)) the most challenging. Rigid object replacement ([Edit1](#)) and object addition ([Edit5](#)) are relatively simple, while non-rigid transformations ([Edit2](#)) and material changes ([Edit4](#)) pose moderate challenges. The particularly low success rate of object removal highlights its complexity, requiring precise inpainting and temporal consistency.

## G. More Qualitative Results and Analysis

We present the editing results across various editing types and comparison methods in Figs. 10 - 13. Specifically, Figs. 10 - 12 compare the results of diffusion-based and RF-based editing methods across six editing types. These comparisons highlight the strengths and weaknesses of each method while also demonstrating the superiority of VideoGrain and our Wan-Edit. The videos shown in Fig. 13 are generated examples and represent a particularly challenging editing case. Editing becomes difficult when the object occupies a significant portion of the video frame, as it requires modifying low-frequency noise while maintaining spatial and temporal consistency. This results in failure even for the best-performing Wan-Edit. To better showcase the dynamic consistency in video editing, more video demos are available on the website: <https://sites.google.com/view/five-benchmark>.

## References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18392–18402, 2023. 2
- [2] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *Int. Conf. Learn. Represent.*, 2024. 1, 8, 9, 10
- [3] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman,

Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 3

- [4] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhua Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. 2, 8, 9, 10
- [5] Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. Vidtope: Video token merging for zero-shot video editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7486–7495, 2024. 1, 8, 9, 10
- [6] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 3
- [7] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1921–1930, 2023. 2
- [8] Xiangpeng Yang, Linchao Zhu, Hehe Fan, and Yi Yang. Videograin: Modulating space-time attention for multi-grained video editing. *arXiv preprint arXiv:2502.17258*, 2025. 2, 8, 9, 10
- [9] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8466–8476, 2024. 1, 8, 9, 10

Table 2. **Edit1**: Comparison of diffusion- and flow-based video editing methods for **object replacement without non-rigid transformations** on the FiVE benchmark.

Methods	Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.	
	Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS.-edit $\uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$	
	Source Videos	0.00	$\infty$	0.00	0.00	100.00	23.97	18.75	6.33	93.76
DMs	TokenFlow [2]	36.24	19.12	257.32	137.12	72.63	27.04	21.23	<b>4.05</b>	88.23
	DMT [9]	85.35	14.35	410.83	402.43	51.61	27.26	<b>21.52</b>	5.25	81.93
	VidToMe [5]	23.81	21.32	261.19	88.47	71.50	<b>27.75</b>	21.19	4.68	<b>90.65</b>
	AnyV2V [4]	71.19	15.85	345.77	350.37	50.91	25.41	19.96	4.64	61.63
	VideoGrain [8]	<b>11.71</b>	<b>26.92</b>	<u>184.08</u>	<b>26.82</b>	<u>79.06</u>	<u>27.43</u>	<u>21.40</u>	<u>4.10</u>	88.29
RFs	Pyramid-Edit	28.27	20.87	276.18	96.15	72.56	<u>27.43</u>	20.11	5.47	81.52
(Ours)	Wan-Edit	<u>13.50</u>	<u>24.81</u>	<b>93.67</b>	<u>39.67</u>	<b>82.54</b>	27.19	21.38	6.59	<u>89.37</u>

Table 3. **Edit2**: Comparison of diffusion- and flow-based video editing methods for **object replacement with non-rigid transformations** on the FiVE benchmark.

Methods		Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.
		Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS.-edit $\uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$
Source Videos		0.00	$\infty$	0.00	0.00	100.00	22.13	17.33	6.33	93.76
DMs	TokenFlow [2]	38.88	19.08	246.65	138.69	72.99	26.30	19.70	<u>4.21</u>	<u>88.19</u>
	DMT [9]	91.04	14.26	413.98	412.38	50.37	<u>26.86</u>	<b>20.38</b>	5.22	80.23
	VidToMe [5]	27.95	20.80	264.37	95.43	70.98	<b>26.95</b>	19.94	4.82	<b>88.97</b>
	AnyV2V [4]	70.14	16.01	350.89	325.25	49.66	23.87	18.46	4.62	60.44
	VideoGrain [8]	<b>11.24</b>	<b>27.23</b>	<u>180.61</u>	<b>26.13</b>	<u>79.56</u>	25.46	19.35	<b>4.12</b>	87.38
RFs	Pyramid-Edit	30.00	20.65	279.11	101.41	71.74	<u>26.86</u>	18.97	5.52	79.55
(Ours)	Wan-Edit	<u>14.33</u>	<u>24.54</u>	<b>96.53</b>	<u>40.36</u>	<b>82.33</b>	26.85	<u>19.98</u>	6.63	87.94

Table 4. **Edit1**: Comparison of diffusion- and flow-based video editing methods for **object replacement without non-rigid transformations** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	30.30	48.48	49.49	29.29	39.39
DMT [9]	<b>55.95</b>	<b>79.76</b>	<b>79.76</b>	<b>55.95</b>	<b>67.86</b>
VidToMe [5]	25.25	47.47	47.47	25.25	36.36
AnyV2V [4]	27.27	42.42	44.44	25.25	34.85
VideoGrain [8]	40.00	62.00	62.00	40.00	51.00
Pyramid-Edit	27.27	53.54	55.56	25.25	40.40
Wan-Edit	<u>41.41</u>	<u>62.63</u>	<u>62.63</u>	<u>41.41</u>	<u>52.02</u>

Table 5. **Edit2**: Comparison of diffusion- and flow-based video editing methods for **object replacement with non-rigid transformations** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	18.18	37.37	38.38	17.17	27.78
DMT [9]	<b>41.49</b>	<u>65.96</u>	<u>68.09</u>	<b>39.36</b>	<b>53.72</b>
VidToMe [5]	23.23	41.41	43.43	21.21	32.32
AnyV2V [4]	13.13	26.26	30.30	9.09	19.70
VideoGrain [8]	12.24	26.53	26.53	12.24	19.39
Pyramid-Edit	30.30	60.61	60.61	30.30	45.45
Wan-Edit	<u>36.36</u>	<b>67.68</b>	<b>68.69</b>	<u>35.35</u>	<u>52.02</u>

Table 6. **Edit3**: Comparison of diffusion- and flow-based video editing methods for **object color changes** on the FiVE benchmark.

Methods	Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.	
	Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS. $\text{-edit} \uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$	
Source Videos	0.00	$\infty$	0.00	0.00	100.00	26.12	20.64	6.33	93.76	
TokenFlow [2]	35.03	19.07	262.05	138.21	72.71	27.72	21.68	<b>4.02</b>	88.37	
DMT [9]	85.75	14.23	413.08	413.91	50.68	28.18	22.10	5.20	81.71	
DMs VidToMe [5]	21.90	21.17	261.94	89.81	72.20	<b>28.45</b>	<u>22.16</u>	4.68	<b>89.26</b>	
AnyV2V [4]	79.16	14.37	411.83	455.53	46.82	26.58	20.59	4.65	61.13	
VideoGrain [8]	<u>14.20</u>	<b>27.08</b>	<u>185.38</u>	<b>25.95</b>	<u>79.37</u>	<u>28.44</u>	<b>22.23</b>	<u>4.09</u>	87.41	
RFs Pyramid-Edit	29.37	20.85	278.17	96.16	72.11	28.10	21.13	5.49	78.80	
(Ours) Wan-Edit	<b>11.63</b>	<u>25.32</u>	<b>90.82</b>	<u>35.77</u>	<b>83.04</b>	27.39	22.04	6.58	<u>88.59</u>	

Table 7. **Edit4**: Comparison of diffusion- and flow-based video editing methods for **object material changes** on the FiVE benchmark.

Methods	Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.	
	Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS. $\text{-edit} \uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$	
Source Videos	0.00	$\infty$	0.00	0.00	100.00	26.89	21.85	6.33	93.76	
TokenFlow [2]	35.22	19.22	258.53	133.83	73.06	27.67	22.19	<u>4.10</u>	88.34	
DMT [9]	84.34	14.15	408.88	417.24	50.24	27.33	<u>22.28</u>	5.14	80.47	
DMs VidToMe [5]	23.19	20.71	273.64	98.70	69.49	<u>27.82</u>	21.91	4.70	<b>89.24</b>	
AnyV2V [4]	71.97	15.53	354.39	382.99	50.33	26.42	20.99	4.60	62.14	
VideoGrain [8]	<b>10.34</b>	<b>27.21</b>	<u>185.78</u>	<b>25.61</b>	<u>79.13</u>	27.44	21.15	<b>4.02</b>	87.55	
RFs Pyramid-Edit	28.39	20.74	277.73	98.54	72.04	<b>28.07</b>	21.48	5.44	78.27	
(Ours) Wan-Edit	<u>10.66</u>	<u>25.45</u>	<b>89.72</b>	<u>34.19</u>	<b>83.31</b>	27.55	<b>22.35</b>	6.57	<u>88.83</u>	

Table 8. **Edit3**: Comparison of diffusion- and flow-based video editing methods for **object color changes** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	34.34	46.46	48.48	32.32	40.40
DMT [9]	55.06	61.80	64.04	52.81	58.43
VidToMe [5]	36.36	42.42	43.43	35.35	39.39
AnyV2V [4]	54.55	<u>64.65</u>	67.68	51.52	59.60
VideoGrain [8]	<b>82.00</b>	<b>90.00</b>	<b>92.00</b>	<b>80.00</b>	<b>86.00</b>
Pyramid-Edit	59.60	57.58	66.67	50.51	58.59
Wan-Edit	<u>62.63</u>	63.64	<u>68.69</u>	<u>57.58</u>	<u>63.13</u>

Table 9. **Edit4**: Comparison of diffusion- and flow-based video editing methods for **object material changes** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	11.11	26.26	29.29	8.08	18.69
DMT [9]	11.76	47.06	48.24	10.59	29.41
VidToMe [5]	13.13	36.36	38.38	11.11	24.75
AnyV2V [4]	<u>23.23</u>	<b>63.64</b>	<b>64.65</b>	<u>22.22</u>	43.43
VideoGrain [8]	<b>26.53</b>	40.82	40.82	<b>26.53</b>	33.67
Pyramid-Edit	18.18	<u>54.55</u>	<u>57.58</u>	15.15	<u>36.36</u>
Wan-Edit	19.19	<b>43.43</b>	45.45	17.17	31.31

Table 10. **Edit5**: Comparison of diffusion- and flow-based video editing methods for **object addition** on the FiVE benchmark.

Methods	Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.
	Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS.-edit $\uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$
Source Videos	0.00	$\infty$	0.00	0.00	100.00	23.52	19.58	5.58	97.86
TokenFlow [2]	36.76	18.55	295.59	152.12	67.04	<b>25.32</b>	<u>21.07</u>	<b>3.25</b>	96.71
DMs DMT [9]	93.29	15.82	413.62	275.22	47.90	25.14	20.85	5.14	91.70
VidToMe [5]	<u>22.31</u>	20.55	275.19	<u>93.10</u>	64.19	24.38	19.73	4.31	97.13
AnyV2V [4]	55.00	16.68	328.01	249.76	48.49	25.01	20.40	<u>4.10</u>	62.76
VideoGrain [8]	<b>18.03</b>	<b>26.42</b>	<u>208.18</u>	<b>27.36</b>	<b>74.44</b>	22.30	18.52	<u>4.10</u>	<b>98.36</b>
RFs Pyramid-Edit	29.24	20.33	293.97	100.99	65.23	24.83	20.25	5.39	89.90
(Ours) Wan-Edit	23.04	<u>20.70</u>	<b>139.79</b>	93.43	<u>73.55</u>	25.09	<b>21.32</b>	5.84	<u>97.38</u>

Table 11. **Edit6**: Comparison of diffusion- and flow-based video editing methods for **object removal** on the FiVE benchmark.

Methods	Structure	Background Preservation				Text Alignment		IQA	Temp. Consis.
	Dist. $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	CLIPS. $\uparrow$	CLIPS.-edit $\uparrow$	NIQE $\downarrow$	Motion Fidelity S. $\times 10^2 \uparrow$
Source Videos	0.00	$\infty$	0.00	0.00	100.00	24.91	21.06	6.79	92.63
TokenFlow [2]	31.61	19.33	261.55	131.90	76.63	24.70	21.02	<u>4.42</u>	84.15
DMs DMT [9]	75.91	15.43	367.21	315.51	59.05	25.22	<b>21.53</b>	5.49	77.76
VidToMe [5]	15.03	22.35	247.12	66.99	75.80	<b>25.67</b>	<u>21.34</u>	4.91	<b>85.12</b>
AnyV2V [4]	80.71	16.97	300.65	293.93	58.38	22.06	17.93	4.93	54.07
VideoGrain [8]	<u>8.86</u>	<u>27.46</u>	<u>167.21</u>	<u>18.76</u>	<u>83.23</u>	23.05	19.22	<b>4.06</b>	82.46
RFs Pyramid-Edit	26.63	21.58	254.41	80.54	76.65	<u>25.60</u>	19.26	5.55	75.52
(Ours) Wan-Edit	<b>2.02</b>	<b>32.62</b>	<b>57.12</b>	<b>7.63</b>	<b>90.52</b>	24.29	20.31	7.02	<u>84.50</u>

Table 12. **Edit5**: Comparison of diffusion- and flow-based video editing methods for **object addition** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	22.22	44.44	44.44	22.22	33.33
DMT [9]	44.44	<b>77.78</b>	<u>77.78</u>	44.44	61.11
VidToMe [5]	22.22	33.33	44.44	11.11	27.78
AnyV2V [4]	55.56	<u>55.56</u>	66.67	44.44	55.56
VideoGrain [8]	22.22	33.33	33.33	22.22	27.78
Pyramid-Edit	<u>66.67</u>	<b>77.78</b>	<u>77.78</u>	<u>66.67</u>	<u>72.22</u>
Wan-Edit	<b>88.89</b>	<b>77.78</b>	<b>88.89</b>	<b>77.78</b>	<b>83.33</b>

Table 13. **Edit6**: Comparison of diffusion- and flow-based video editing methods **object removal** on the FiVE benchmark using FiVE-Acc metrics.

Method	YN-Acc	MC-Acc	U-Acc	$\cap$ -Acc	FiVE-Acc $\uparrow$
TokenFlow [2]	0.00	10.00	10.00	0.00	5.00
DMT [9]	0.00	<b>40.00</b>	<b>40.00</b>	0.00	<b>20.00</b>
VidToMe [5]	0.00	0.00	0.00	0.00	0.00
AnyV2V [4]	<b>10.00</b>	<u>20.00</u>	<u>20.00</u>	<b>10.00</b>	<u>15.00</u>
VideoGrain [8]	0.00	11.11	11.11	0.00	5.56
Pyramid-Edit	0.00	<u>20.00</u>	<u>20.00</u>	0.00	10.00
Wan-Edit	0.00	0.00	0.00	0.00	0.00

*Edit1 Object (w/o non-rigid deform): Dog → Rabbit*



*Edit2 Object (w non-rigid deform): A young girl → A young alien*



*Edit3 Color: A gray dog → A pink dog*



*Edit4 Material: A wheelchair → A wooden wheelchair*



*Edit6 Remove: A young girl ..., not accompanied by a gray dog walking alongside her.*



Source video

TokenFlow

DMT

VidToMe

VideoGrain

Pyramid-Edit

Wan-Edit

Figure 10. Editing results across five editing types and six high-performance comparison methods.

Edit1 Object (w/o non-rigid deform): Bear → Panda



Edit2 Object (w non-rigid deform): Bear → Dinosaur



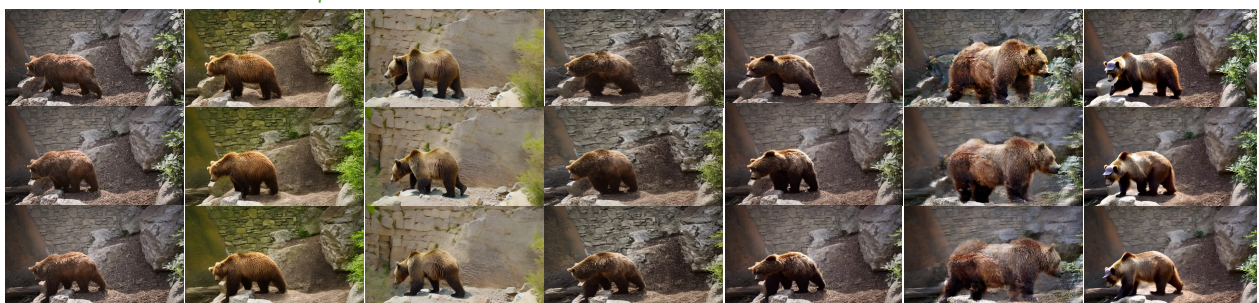
Edit3 Color: A bear → A purple bear



Edit4 Material: A bear → A bronze bear



Edit5 Add: A bear → A bear with cap



Source video

TokenFlow

DMT

VidToMe

VideoGrain

Pyramid-Edit

Wan-Edit

Figure 11. Editing results across five editing types and six high-performance comparison methods. Wan-Edit is the only method that succeeds in the object addition editing type.

*Edit1 Object (w/o non-rigid deform): A tennis player → A ultraman*



*Edit2 Object (w non-rigid deform): A tennis player → A robot*



*Edit3 Color: White shorts → Black shorts*



*Edit4 Material: A tennis player → A clay tennis player*



*Edits Add: A tennis player wearing a bright yellow fedora*



Source video

TokenFlow

DMT

VidToMe

VideoGrain

Pyramid-Edit

Wan-Edit

Figure 12. Editing results across five editing types and six high-performance comparison methods.

Generated video:  
A cyclist wearing a  
helmet is pedaling  
vigorously ...

Edit1:  
A skateboarder

Edit2:  
A rollerblader  
cyclist

Edit3:  
Cardboard cyclist

Edit4:  
A maroon cyclist

Edit6:  
Not wearing a  
helmet



Figure 13. A generated video (first row) along with Wan-Edit's editing results across five editing types (rows 2-6).