CVPR #10991

OmniInsert: Mask-Free Video Insertion of Any Reference via Diffusion Transformer Models

Supplementary Material

In the supplementary material, the sections are organized as follows:

- We provide more details regarding parameters in Sec. A.
- We provide more details of our proposed benchmark ***InsertBench*** in Sec. B.
- We provide more qualitative results and comparisons of ***OmniInsert*** in Sec. C.
- We provide an additional comparison with the mask-based video insertion methods in Sec. D.
- We provide more details of our proposed Context-Aware Rephraser (CAR) in Sec. E.
- We discuss the limitations and the future work of our method in Sec. F.

A. Implementation Details

A.1. Detailed Parameters

The hyper-parameters used in our experiments are set as follows:

- For the $\mathcal{L}_{\text{phase1-3}}$ mentioned in Eq. 5, we set λ_1 and λ_2 as 1, 1, respectively.
- For the $\mathcal{L}_{\text{reg}}(c, x_w, x_l)$ mentioned in Eq. 7, we set the hyper-parameters β to 5.

A.2. Generalizability of the Method

As discussed in the main article, our method is built upon a vanilla I2V base model. By reusing and modifying the model's native input channels and training LoRA modules without incorporating auxiliary structures, we equip the model with an end-to-end video insertion capability. Theoretically, this design allows our approach to adapt to the majority of I2V architectures. We have validated the effectiveness of our method on both the open-source Wan14B-I2V [10] model and an in-house I2V model, the latter of which yields superior performance. All qualitative and quantitative results, as well as the extended discussions and analyses presented in this paper, are based on this in-house model. We plan to open-source the model checkpoint based on Wan14B-I2V after proper organization and cleaning.

A.3. Training and Inference Details

Training time. The phase-1 subject-to-video training takes about 70k iterations (nearly 3,600 A100 GPU hours). The phase-2 pretraining for the mask-free video insertion task costs about 30k iterations (nearly 1,800 A100 GPU hours), and the phase-3 refinement costs about 10k iterations (nearly 600 A100 GPU hours). The last phase-4 preference optimization takes about 8k iterations (nearly 1,200 A100 GPU hours).

Trainable Parameters. We integrate the LoRA mechanism into the DiT blocks, configuring it with a rank of 256. This results in a total of 0.6B trainable parameters. A comparison between ours and existing methods refers to Tab. 4. Notably, the following methods are excluded from the training data comparison: (1) training-free approaches, such as [4, 13] which are based on DDIM inversion; and (2) methods which are closed-source and do not disclose the amount of trainable parameters in their papers or technical reports (e.g., the academic study like [6, 9, 12], and commercial models like [3, 7]). For works that provide multiple versions (e.g., [2]), we use the amount of trainable parameters from the best-performing version for the comparison. Since most existing methods introduce computationally heavy structures (e.g., ControlNet) or require full-parameter fine-tuning of the base model, whereas our approach only trains highly parameter-efficient LoRA modules, our method demonstrates significantly fewer trainable parameters compared to prior works.

Table 4. The trainable parameters of existing methods and ours.

Methods	Trainable Parameters
VACE [2]	∼3B
GetInVideo [14]	2B
Ours	∼0.6B

Training Data Details. For the training of the phase-1 subject-to-video task and the phase-2 video insertion task, we use a large-scale training dataset containing about 0.5M samples. Mark data generated by the introduced RealCapture Pipe, SynthGen Pipe (T2I + I2V + subject removal), SynthGen Pipe (video editing), and SimInteract Pipe as **(a)**, **(b)**, **(c)**, **(d)**, respectively. The actual ratio of different types of data (type **(a)** : type **(b)** : type **(c)** : type **(d)**) in phase 1 and phase 2 is set to 5:2:2:1. For the training of phase-3 refinement, the dataset contains about 50k samples and the ratio of different data is 3:3:3:1. For the training of phase-4 preference optimization, the training dataset has about 0.5k good-bad paired data, and the ratio is set to 3:3:3:1. A comparison about the amount of the training data between ours and existing methods refers to Tab. 5. Similarly, the following methods are excluded from the training data comparison: (1) training-free approaches, such as [4, 13] which are

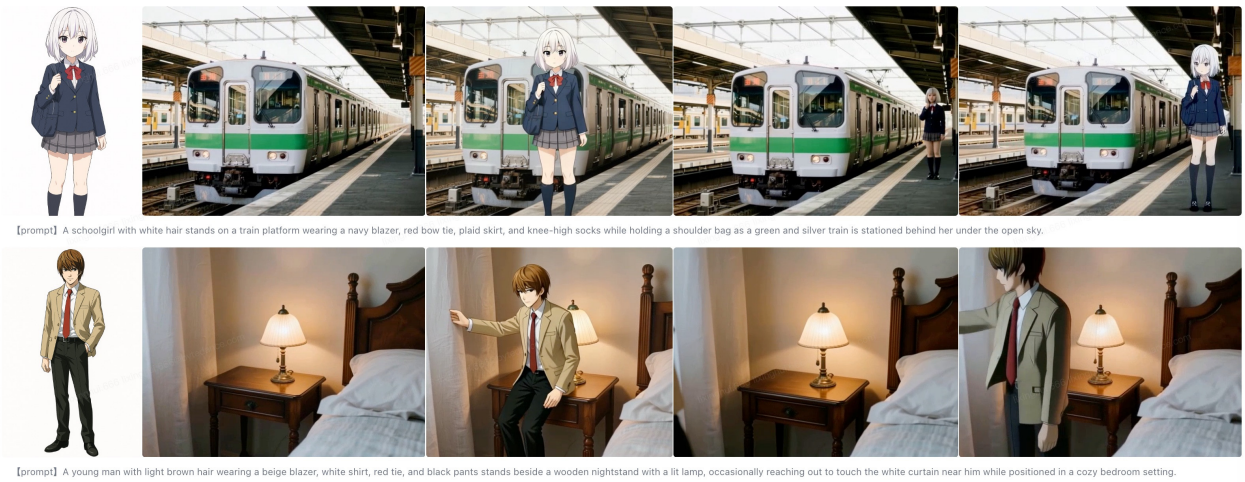

Figure 7. Screenshot of the **user study**.

based on DDIM inversion; and (2) methods that do not disclose the amount of training data in their papers or technical reports (e.g., the academic study like [2, 6], and commercial models like [3, 7]). For VideoAnydoor [9], it chooses the frame that has the largest distance from the picked target video clip as the reference image; For GetInVideo [14], it selects the first clear frame containing the reference subject from the target video as the reference image; For Unic [12], it directly uses the part of target video as the reference. Our training dataset is the **first** cross-paired dataset constructed for the video insertion task, which organizes the reference image from a brand new source (cross-paired) instead of the target video itself (in-paired). Our dataset will be open-sourced after proper organization and cleaning.

Table 5. The training data of existing methods and ours.

Methods	Training Data	Source Video	Cross-paired
VideoAnydoor [9]	∼440K	✗	✗
GetInVideo [14]	1,000K	✓	✗
Unic [12]	∼166K	✓	✗
Ours: Phase1-2	500K	✓	✓
Ours: Phase3	50K	✓	✓
Ours: Phase4	0.5K	✓	✓
Ours: Total	∼551K	✓	✓

Inference Details of the Model. Classifier-free guidance [1] balances sample quality and diversity in diffusion models through joint conditional and unconditional training. During inference, we design a joint classifier-free guidance to balance multiple conditions:

$$\begin{aligned}\hat{\mathbf{v}}_\theta(\mathbf{z}_t, \mathbf{c}_p, \mathbf{c}_i, \mathbf{c}_v) = \mathbf{v}_\theta(\mathbf{z}_t, \emptyset, \emptyset, \emptyset) \\ + S_1 \cdot (\mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}_p, \emptyset, \emptyset) - \mathbf{v}_\theta(\mathbf{z}_t, \emptyset, \emptyset, \emptyset)) \\ + S_2 \cdot (\mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}_p, \mathbf{c}_i, \emptyset) - \mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}_p, \emptyset, \emptyset)) \\ + S_3 \cdot (\mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}_p, \mathbf{c}_i, \mathbf{c}_v) - \mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}_p, \mathbf{c}_i, \emptyset)),\end{aligned} \tag{8}$$

where $\mathbf{c}_p$, $\mathbf{c}_i$, $\mathbf{c}_v$ represent prompt condition, reference subjects, and source video. S_1, S_2 and S_3 are guidance scales, which are assigned as 3.5, 5.0, 1.5, respectively. We run a standard Flow Euler sampler with 50 steps to obtain our inference results.

Inference time. Generating a single 5-second 480P video (121 frames) with our proposed model takes approximately 45 seconds using 8 NVIDIA A100 GPUs. Under these conditions, a further comparison between ours and existing methods refers to Tab. 6. Note that inversion-based methods [4, 13] are excluded from our comparison due to their additional inversion stage introducing significant computational overhead. For works that provide multiple versions (e.g., [2]), we use the inference time from the best-performing version for the comparison. We exclude closed-source methods without publicly accessible demos from our comparison. For closed-source methods with available demos (e.g., Kling [3] and Pika-Pro [7]), we record their inference time in our tables. However, these results are marked with * and should be interpreted cautiously, as the actual computational resources used by these services cannot be verified. Further discussion on potential improvements to inference speed can be found in Sec. F.

A.4. User Study

To compare with the baseline methods, we conduct a user study as part of the evaluation. The survey randomly presented 40 sets of generated results to each participant. Fig. 7

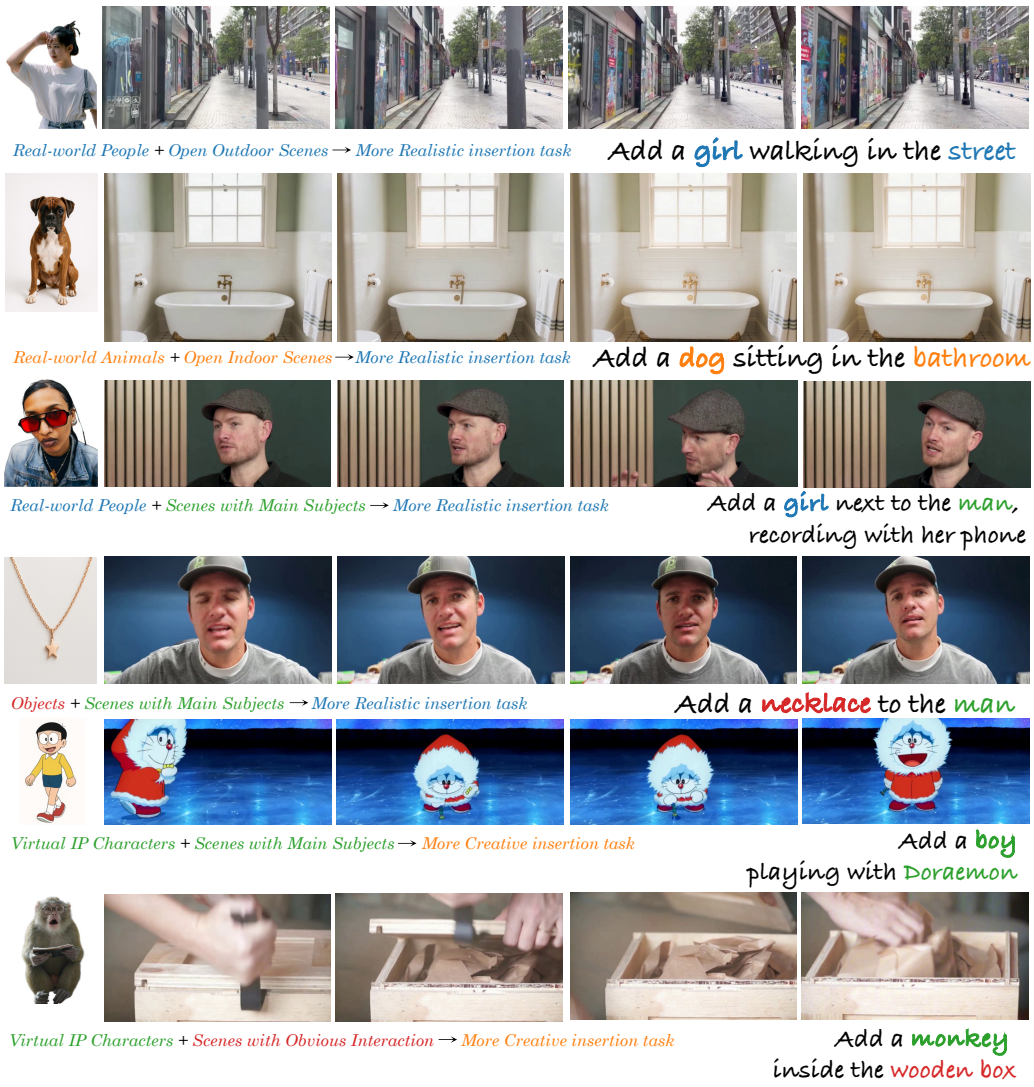

Figure 8. Examples of ***InsertBench***.

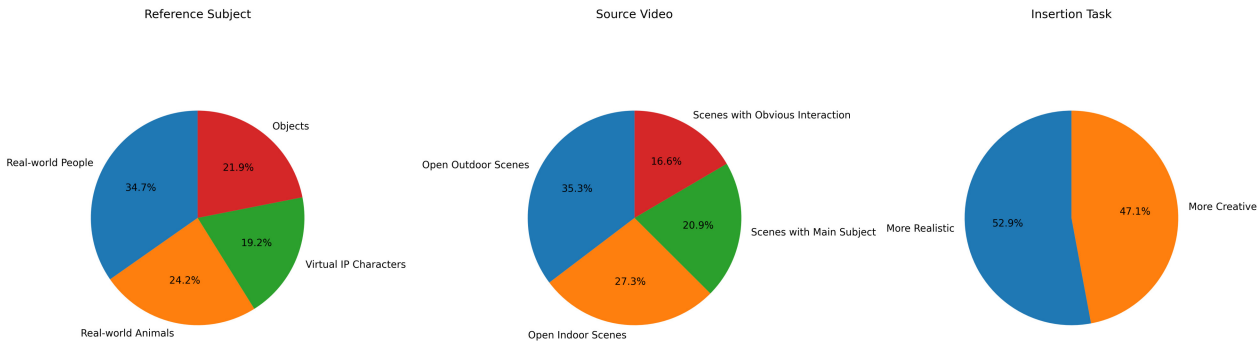

Figure 9. The distribution of ***InsertBench***.

displays a screenshot from our user study, showcasing a set of generated results. From left to right, it shows the reference subject, the source video, and the results from all competing methods in random order, paired with the corre-

Table 6. The inference time of existing methods and ours.

Methods	Inference Time
VACE [2]	∼180s
Kling [3]	∼120s*
Pika-Pro [7]	∼150s*
Ours:	∼45s

sponding text prompt. We ask each participant four questions:

1. *Which result appears to have the highest consistency with reference subjects?*
2. *Which result best matches the prompt '[prompt]'?*
3. *Which result appears to have the highest insertion rationality?*
4. *Which result matches your best choice based on comprehensive considerations?*

For each set of results displayed in the survey, we ensured that their order was randomly shuffled to prevent bias. Responses where all answers had the same selection and responses with completely identical answers were considered invalid.

B. InsertBench Details

To address the lack of a benchmark for video insertion tasks, we introduce a comprehensive benchmark, ***InsertBench***, which consists of 170 videos paired with meticulously selected corresponding subjects (suitable for insertion in each video) and the prompts. Our collected videos span diverse categories, including natural landscapes, indoor environments, transportation scenes, dynamic interactions, animated contexts and wearable scenarios, each comprising 121 frames at 24 fps. As shown in Fig. 8, we carefully select suitable subjects for each video to insert, while considering harmonization and creativity. The detailed data type distribution of our benchmark is shown in Fig. 9. A current limitation in our benchmark is that there are insufficient source videos containing prominent interactive actions. Such videos are critically important for both generating creative effects and validating models' understanding of physical laws. Consequently, a key future improvement direction for this benchmark involves augmenting the collection with more interaction-oriented source videos. Additionally, we will incorporate statistical analysis and evaluation of video styles (e.g., real-world style, cartoon style) in subsequent versions.

C. More Visual Results

In this section, we provide more visual results of ***OmniInsert***. Fig. 11 and Fig. 12 present our inference results and comparisons with baselines. Corresponding videos for all results (main paper and supplement) are available in supplementary attachments. **We strongly recommend viewing the mentioned dynamic results for a more intuitive understanding of the practical effectiveness of our proposed method.**

All results are generated at 480P resolution based on our proposed benchmark *InsertBench*. Supplemental results demonstrate enhanced capabilities in diverse aspect ratios (e.g., vertical video) and multi-subject scenarios, further demonstrating that our method can achieve stable, semantically coherent insertions with strong prompt fidelity. Thanks to our designed data curation pipeline *InsertPipe*, which covers a wide range of scenes and subject categories, the model exhibits excellent robustness across various resolutions and aspect ratios. Our designed Condition-Specific Feature Injection (CFI) mechanism injects both video and subject features in a unified yet efficient manner, ensuring differentiated condition injection. And we integrate the LoRA mechanism into the DiT blocks, preserving the model's original prompt following capabilities. Furthermore, the adopted progressive training strategy enables the model to effectively achieve subject-scene equilibrium through multi-stage optimization, further enhancing insertion stability. As shown in Fig. 12, both Pika-Pro [7] and Kling [3] exhibit issues with insertion failures and unnatural insertion artifacts, whereas our method demonstrates superior robustness.

Additionally, our model is capable of dealing with multi-subject scenarios, a capability that other baseline models either lack or perform poorly on. Our proposed CFI can be effortlessly extended from a single-reference to a multi-reference setup. This is achieved by simply concatenating all reference images in the temporal order. This approach introduces no additional trainable parameters and requires no modifications to the training paradigm. The corresponding results are visualized at the bottom of Fig. 1 and Fig. 11, and the associated original files are included in the attachment of our supplementary material.

D. Additional comparison

In this section, we compare our method with another type of approaches: **mask-based** video insertion methods. These methods, such as VACE [2], rely heavily on a user-provided mask to guide the placement of the inserted object. This requirement has two significant drawbacks: first, creating the costly mask is not user-friendly; second, it inherently constrains the model's creative potential. In contrast, our proposed mask-free method is more intuitive to use and more conducive to generating imaginative results. Qualitative comparisons are visualized in Fig. 13, while quantitative results from automated metrics are presented in Tab. 7. As evidenced by both qualitative and quantitative evalua-

CVPR #10991

Table 7. **Quantitative comparisons** with mask-based method, and * indicates the consistency of the segmented subject area.

Method	Subject Consistency			Text-Video Alignment	Video Quality			
	CLIP-I* ↑	DINO-I* ↑	FaceSim ↑	ViCLIP-T ↑	Dynamic ↑	Image-Quality ↑	Aesthetics ↑	Consistency ↑
VACE [2]	0.673	0.532	0.348	24.860	0.731	0.653	0.543	0.929
Ours	**0.745**	**0.639**	**0.488**	**25.945**	**0.825**	**0.704**	**0.556**	**0.930**

tions, our method outperforms VACE across several key dimensions, including reference consistency, text-prompt fidelity, and physical plausibility.

E. Context-Aware Rephraser (CAR)

We employ CAR during the inference stage to either elaborate on concise user-provided prompts or, in their absence, generate a semantically rich prompt from scratch. CAR accepts three inputs: source video, reference image, and an optional prompt. The module is powered by a pre-trained Vision-Language Model (VLM); specifically, we use Doubao VLM [8] as the backend. The user-provided prompt can be divided into three categories: imperative, descriptive, or null (empty). Given that our main model is trained on descriptive-style prompts, CAR is engineered to convert all these three input types into a detailed descriptive prompt. This conversion process is conditioned on the provided source video and reference image, and the final output serves as the text condition for our model. Notably, we use 20 uniformly sampled frames as a substitute for the full source video when feeding it into CAR. The complete instructions for CAR, including the system and user prompts, will be open-sourced with our code implementation. The text generation process of CAR is illustrated in Fig. 14.

F. Limitation and Future Work

In this section, we discuss the limitations of our method and potential directions for future work.

Limitations Imposed by the Base Model. Our method is built on a naive I2V base model. By exclusively training LoRA modules without altering the model's core architecture, we enable it to generalize from the I2V modality to the V2V capability, ultimately accomplishing the video insertion task. The primary advantage of this approach is that it maximally preserves the base model's high-quality pre-trained abilities (e.g., generated quality and prompt fidelity) while incurring significantly lower computational costs than full-parameter fine-tuning or incorporating auxiliary structures like ControlNet. On the other hand, since we do not modify the base model's intrinsic architecture or parameters, the upper bound of our performance is inherently constrained by the base model's own capability domain (e.g., maximum generation resolution and count of the generated frames). We observe that when running inference on cases with out-of-domain parameters, such as higher resolutions or longer durations, the model frequently generates physically implausible results shown in Fig. 10 (a) (the same case as that shown in Fig. 1 but with a longer duration of 18s), and occasionally produces outputs with significant color discrepancies from the source video shown in Fig. 10 (b) (inference with a larger source video of 1080p). Although our dataset is diverse in parameters like duration and resolution, and our multi-stage progressive training strategy is theoretically designed to mitigate these problems, we still find that the base model's inherent characteristics **persistently** impact performance during out-of-domain-parameter inference. Therefore, a foreseeable direction for future work is to strike a better trade-off between training efficiency and model performance. This involves exploring how to optimize the method with more suitable trainable architectures and a larger set of trainable parameters, thereby reducing the dependency on the base model's inherent limitations.

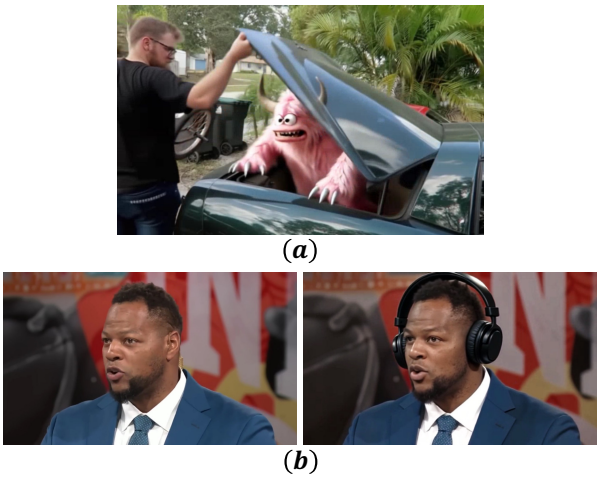

Figure 10. The failure cases. **(a)** shows the physically implausible bad cases, such as model interpenetration, and **(b)** shows the slight color discrepancy between the source video and the inference result.

Inference Speed. Our method adheres to a standard video diffusion model baseline, with an inference time of approximately 45 seconds for a 5-second 480P video on 8 A100 GPUs. Its performance could be further enhanced by applying general-purpose acceleration techniques [5, 11] for video diffusion models.

References

[1] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[2] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 1, 2, 4, 5

[3] Keling. Image to video elements feature. https://app.klingai.com/cn/multimodal-to-video/add-object/new, 2025. 1, 2, 4

[4] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. 1, 2

[5] Feng Liang, Akio Kodaira, Chenfeng Xu, Masayoshi Tomizuka, Kurt Keutzer, and Diana Marculescu. Looking backward: Streaming video-to-video translation with feature banks. *arXiv preprint arXiv:2405.15757*, 2024. 5

[6] Sen Liang, Zhentao Yu, Zhengguang Zhou, Teng Hu, Hongmei Wang, Yi Chen, Qin Lin, Yuan Zhou, Xin Li, Qinglin Lu, et al. Omniv2v: Versatile video generation and editing via dynamic content manipulation. *arXiv preprint arXiv:2506.01801*, 2025. 1, 2

[7] Pika. Pikaadd. https://pika.art/pikadditions, 2025. 1, 2, 4

[8] ByteDance Seed Team. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025. 5

[9] Yuanpeng Tu, Hao Luo, Xi Chen, Sihui Ji, Xiang Bai, and Hengshuang Zhao. Videoanydoor: High-fidelity video object insertion with precise motion control. *arXiv preprint arXiv:2501.01427*, 2025. 1, 2

[10] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1

[11] Fu-Yun Wang, Zhaoyang Huang, Weikang Bian, Xiaoyu Shi, Keqiang Sun, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Computation-efficient personalized style video generation without personalized video data. In *SIGGRAPH Asia 2024 Technical Communications*, pages 1–5. 2024. 5

[12] Zixuan Ye, Xuanhua He, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Qifeng Chen, and Wenhan Luo. Unic: Unified in-context video editing. *arXiv preprint arXiv:2506.04216*, 2025. 1, 2

[13] Yuyang Zhao, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Make-a-protagonist: Generic video editing with an ensemble of experts. *arXiv preprint arXiv:2305.08850*, 2023. 1, 2

[14] Shaobin Zhuang, Zhipeng Huang, Binxin Yang, Ying Zhang, Fangyikang Wang, Canmiao Fu, Chong Sun, Zheng-Jun Zha, Chen Li, and Yali Wang. Get in video: Add anything you want to the video. *arXiv preprint arXiv:2503.06268*, 2025. 1, 2

Figure 11. More **qualitative results**.

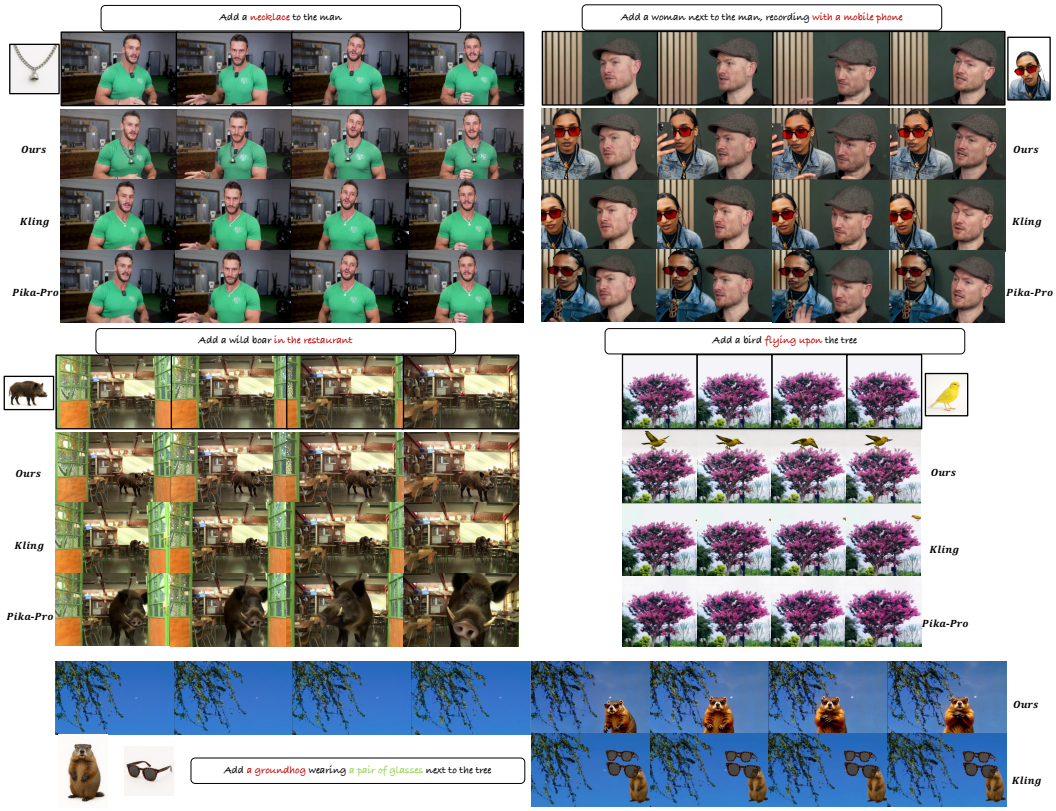

Figure 12. More **qualitative comparisons**.

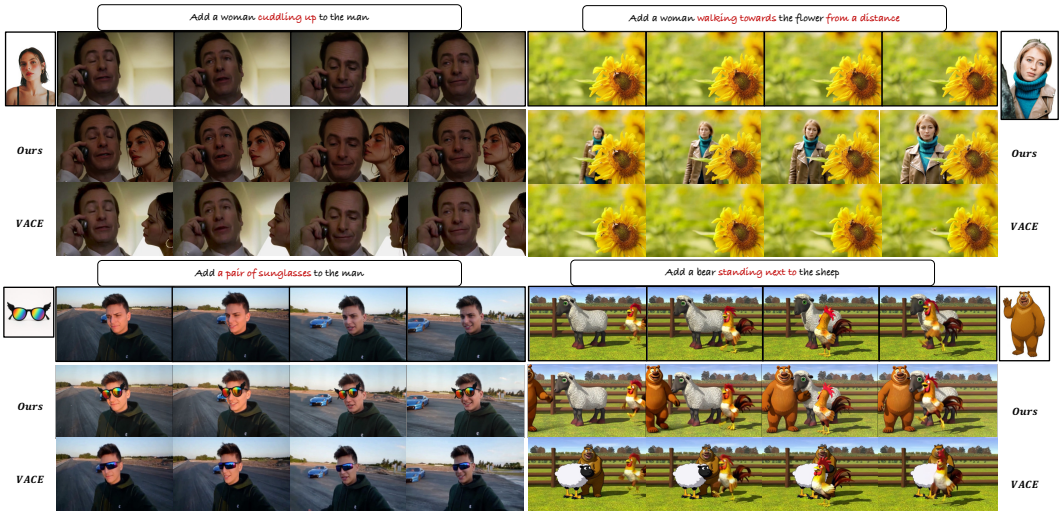

Figure 13. An additional **qualitative comparison** with mask-based methods.

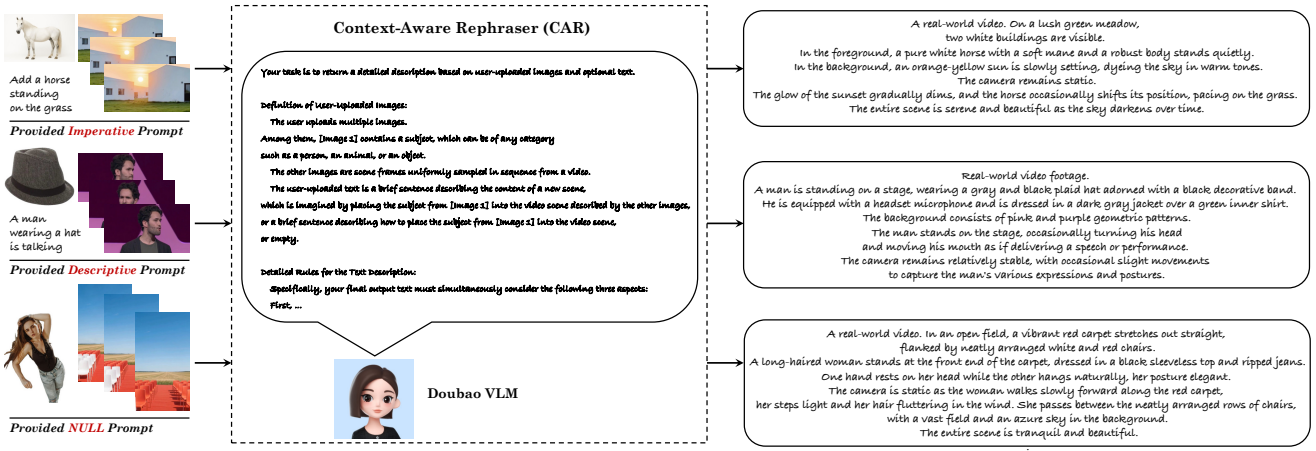

Figure 14. Illustration of **CAR**. Zoom in for more details.