YOSE: You Only Select Essential Tokens for Efficient DiT-based Video Object Removal

Supplementary Material

7. Overview

This supplementary material provides additional technical details and algorithmic descriptions that complement the main paper.

8. Mask Embedding

Diffusion-based video editing frameworks typically rely on a 3D-VAE [20] to encode videos into spatiotemporal latent tokens. However, existing approaches such as VACE [7] directly resize the input mask to the latent resolution, which implicitly assumes a pixel-wise correspondence between the RGB space and the latent space. Since the 3D-VAE encodes the video in spatiotemporal blocks with strides (F_v, H_v, W_v), a simple resize operation does not correctly reflect whether a latent token corresponds to a masked region. As a result, the latent-space mask generated through resizing may incorrectly mark clean regions as masked or miss fine-grained masked areas.

To obtain an accurate latent-space mask that is consistent with the 3D-VAE's block-wise encoding, we introduce a block-wise mask embedding mechanism. Instead of resizing, we determine whether each latent token should be marked as masked by examining the corresponding entire spatiotemporal block in the input mask. Specifically, for every VAE block of size (F_v, H_v, W_v), we aggregate the binary mask values within the block using a multiplicative rule:

$$1 - \prod_{i=0}^{F_v H_v W_v} (1 - m_i), \tag{15}$$

where m_i is the binary mask at the i-th position inside the block. This rule ensures that a latent token is considered masked if and only if any pixel inside its receptive block is masked.

The resulting latent-space mask is fully aligned with the 3D-VAE's tokenization scheme and can seamlessly integrate with our BVI algorithm. Together, they enable precise identification of essential tokens for selective processing in YOSE. The full procedure is summarized in Algorithm 2.

9. Details of DiffSim Module

Algorithm 3 provides the detailed pseudocode of our Diffusion Process Simulator (DiffSim). DiffSim is designed to simulate the diffusion process in DiT while avoiding a full forward pass through all tokens. Given the noise latent Lat_{Nis}, the masked-video latentLat_{mask}, and the corresponding mask $mask$, DiffSim produces the selective latent update used by YOSE.

Algorithm 2 Mask Embedding

Input: $mask \in \mathbb{R}^{B\times 1\times F\times H\times W}$
Output: Emb_{mask}

F_v, H_v, W_v: 3D-VAE Stride
$Num = F_v \times H_v \times W_v$
$mask \overset{reshape}{\Rightarrow}$
  $Emb_{mask} \in \mathbb{R}^{B\times 1\times \frac{F}{F_v}\times F_v\times \frac{H}{H_v}\times H_v\times \frac{W}{W_v}\times W_v}$
$Emb_{mask} \overset{transpose}{\Rightarrow}$
  $Emb_{mask} \in \mathbb{R}^{B\times 1\times \frac{F}{F_v}\times \frac{H}{H_v}\times \frac{W}{W_v}\times F_v\times H_v\times W_v}$
$Emb_{mask} \overset{reshape}{\Rightarrow}$
  $Emb_{mask} \in \mathbb{R}^{B\times 1\times \frac{F}{F_v}\times \frac{H}{H_v}\times \frac{W}{W_v}\times Num}$
$Emb_{mask} = \prod_{i=0}^{Num-1} (1 - Emb_{mask}[..., i])$
return $1 - Emb_{mask}$

Mask–guided token partitioning. We first convert the input mask into a latent-space mask using the Mask Embedding function (Alg. 2). The latent tokens are then divided into foreground (to be updated) and background (to be preserved) subsets using the Batch Variable-length Indexing function ($BIndex(\cdot)$, detailed in Alg. 1). This step yields four index sets. Ind_{F_in}, Ind_{F_out}: indices for tokens fed into the short branch, Ind_{B_in}, Ind_{B_out}: indices for tokens sampled back into the full-resolution latent. This partitioning ensures that only tokens relevant to the edited region will be handled by the DiT-like module.

As shown in Eq. (5), we compute the residual latent Res_{Nis}. Then, using $GSample(\cdot)$, we extract the tokens corresponding to the foreground region, St_{in}.

At each iteration, the current latent features are combined with positional information and passed through a DiT block consisting of attention and feed-forward layers. The attention mechanism incorporates information from the simulated latent contexts, while learnable scaling and bias parameters modulate the intermediate representations to better align them with the statistics of the target domain (St_{in}). Throughout this process, the model repeatedly evaluates which tokens require additional updates based on the mask embedding and the indexing strategy used earlier. Only those latent tokens that correspond to the masked regions are regenerated and refined. Unmasked tokens remain fixed, ensuring stability and avoiding unnecessary computation. The refinement proceeds iteratively until all DiT

blocks have been applied. As a result, the output latent becomes increasingly consistent with both the valid video content and the expected diffusion trajectory, enabling the final reconstruction to better match the underlying spatial-temporal structures of the masked video.

Algorithm 3 Diffusion Process Simulator

Input: Lat_{Noise},Lat_{mask}, $mask$
Output: out

Num_D: The Number of DiT Blocks
$\mathcal{G}$: Learnable Combining Parameters
$\mathcal{S}$: Learnable Scaling Parameters
$\mathcal{B}_{ias}$: Learnable Bias Parameters
Lat_{Nis}: The Input Noise Latent
Lat_{mask}: The Latent of Masked Video
Pos_{emb}: Position Embedding
$BIndex()$: Function of Batch Varible-length Indexing (As Mentioned in Alg. 1)
$GSample()$: Function 'F.grid_sample' in Torch
$Mask\_Emb()$: Function of Mask Embedding
$mask = Mask\_Emb(mask)$
$Ind_{F\_in}, Ind_{B\_in} = BIndex(mask)$
$Ind_{F\_out}, Ind_{B\_out} = BIndex(1 - mask)$
$Res_{Nis} = Lat_{Nis} - Lat_{mask}$
$St_{in} = GSample(Lat_{mask}, Ind_{F\_in})$
for each $i \in [0, Num_D - 1]$ **do**
  $Q = Apply(St_{in}, GSample(Pos_{emb}, Ind_{F\_in}))$
  $KV = \mathcal{G}[i] * Lat_{mask} + (1 - \mathcal{G}[i]) * Res_{Nis}$
  $KV = (1 + \mathcal{S}[i]) * KV + \mathcal{B}_{ias}[i]$
  $KV = Apply(KV \cup St_{in}, Pos_{emb})$
  $St_{in} = Attn\&FFN(Q, K, V)$
end for
$out = mask * GSample(St_{in}, Ind_{B\_in})$
return out

10. More Details

10.1. Causal Encoding of 3D-VAE

Previous study [22] found that the slight localized blurring and color casts in certain cases were primarily artifacts of the 3D-VAE's causal encoding mechanism. This mechanism can introduce statistical inconsistencies at mask boundaries during the latent space transformation, particularly when the content inside the mask is not aligned with its surroundings. To resolve this, YOSE pre-fills masked regions with neighboring pixels prior to encoding, thereby harmonizing feature distributions across boundaries.

10.2. Efficiency Metrics

The Tab. 3 shows a comparison of the speeds of various DiT-based methods. We applied YOSE to VACE, a ControlNet-like DiT-based video editing model. As shown in Fig. 6, the original VACE suffers from mask-shaped semantic bias, inadvertently generating mask-shaped foreground objects, and significantly altering the background. After applying YOSE, we discovered that YOSE's selective token processing effectively suppresses these unwanted hallucinations. By focusing computation exclusively on essential tokens within the mask, YOSE prevents the model from over-interpreting mask semantics, thus achieving the improvement of success rate in object removal(from 62.2% to 97.8%). Since the control branch of VACE consumes significant computation, which cannot be accelerated by YOSE, the acceleration effect of YOSE on this part is relatively limited.

Figure 6. Visual Comparison between VACE and YOSE (VACE).

VideoPainter	ROSE	VACE	YOSE (VACE)	Minimax	YOSE (Minimax)
0.402	1.066	0.308	0.417	9.515	24.509

Table 3. Efficiency Metrics (FPS).

10.3. Relative Performance Change Analysis

As shown in the scatter plot (Fig. 7), we measured the Relative Performance Change (γ /%) across 180 cases in two datasets. 76.1% of cases fall within the mask ratio range of 0–25%, exhibiting minimal fluctuations in visual quality. As the mask ratio increases, the amplitude of fluctuations begins to grow, $\pm 25\%$. Notably, the vast majority of cases, about 75%, fluctuate within a range of $\pm 5\%$, which demonstrates YOSE's high robustness and stability across diverse removal scenarios.

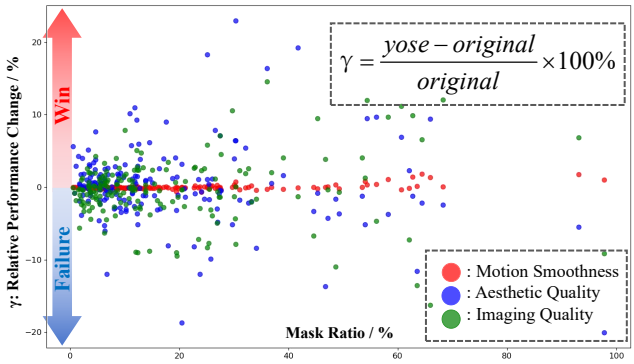

Figure 7. Relative Performance Change Analysis.

References

[1] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *SIGGRAPH*, pages 1–12, 2025. 3, 6

[2] Zheng-Peng Duan, Jiawei Zhang, Xin Jin, Ziheng Zhang, Zheng Xiong, Dongqing Zou, Jimmy S Ren, Chunle Guo, and Chongyi Li. Dit4sr: Taming diffusion transformer for real-world image super-resolution. In *ICCV*, pages 18948–18958, 2025. 3

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 2, 3

[4] Pengcheng Fang, Yuxia Chen, and Rui Guo. When and what: Diffusion-grounded videollm with entity aware segmentation for long video understanding. *CoRR*, abs/2508.15641, 2025. 2, 3

[5] Kunyu Feng, Yue Ma, Bingyuan Wang, Chenyang Qi, Haozhe Chen, Qifeng Chen, and Zeyu Wang. Dit4edit: Diffusion transformer for image editing. In *AAAI*, pages 2969–2977, 2025. 3

[6] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, pages 21807–21818, 2024. 6, 8

[7] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *ICCV*, 2025. 2, 3, 6, 7, 1

[8] Dehong Kong, Fan Li, Zhixin Wang, Jiaqi Xu, Renjing Pei, Wenbo Li, and WenQi Ren. Dual prompting image restoration with diffusion transformers. In *CVPR*, pages 12809–12819, 2025. 2

[9] Fan Li, Zixiao Zhang, Yi Huang, Jianzhuang Liu, Renjing Pei, Bin Shao, and Songcen Xu. Magiceraser: Erasing any objects via semantics-aware control. In *ECCV*, pages 215–231. Springer, 2024. 3

[10] Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025. 6

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[12] Wanglong Lu, Xianta Jiang, Xiaogang Jin, Yong-Liang Yang, Minglun Gong, Kaijie Shi, Tao Wang, and Hanli Zhao. Grig: Data-efficient generative residual image inpainting. *Computational Visual Media*, 11(6):1329–1361, 2025. 3

[13] Yuang Meng, Xin Jin, Lina Lei, Chun-Le Guo, and Chongyi Li. Ultraled: Learning to see everything in ultra-high dynamic range scenes. *arXiv preprint arXiv:2510.07741*, 2025. 3

[14] Chenxuan Miao, Yutong Feng, Jianshu Zeng, Zixiang Gao, Hantang Liu, Yunfeng Yan, Donglian Qi, Xi Chen, Bin Wang, and Hengshuang Zhao. Rose: Remove objects with side effects in videos. *NeurIPS*, 2025. 2, 3, 6

[15] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732. IEEE Computer Society, 2016. 2, 6

[16] Mengjie Qin, Wen Wang, Honghui Xu, Te Li, Chunlong Zhang, and Minhong Wan. Unified transformed t-svd using unfolding tensors for visual inpainting. *Computational Visual Media*, 2025. 3

[17] Xinran Qin, Yuhui Quan, Zhuojie Chen, and Hui Ji. Robust unsupervised deep learning for nonblind image deconvolution with inaccurate kernels. *TNNLS*, 2025. 3

[18] Xinran Qin, Zhixin Wang, Fan Li, Haoyu Chen, Renjing Pei, Wenbo Li, and Xiaochun Cao. Camedit: Continuous camera parameter control for photorealistic image editing. In *NeurIPS*, 2025. 3

[19] Xinran Qin, Yuning Cui, Shangquan Sun, Ruoyu Chen, Wenqi Ren, Alois Knoll, and Xiaochun Cao. Disentangle to fuse: Towards content preservation and cross-modality consistency for multi-modality image fusion. *TIP*, 2026. 3

[20] 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. 2, 3, 1

[21] Chenyang Wu, Jiayi Fu, Chun-Le Guo, Shuhao Han, and Chongyi Li. Vtinker: Guided flow upsampling and texture mapping for high-resolution video frame interpolation. *arXiv preprint arXiv:2511.16124*, 2025. 2

[22] Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. In *CVPR*, pages 18124–18133, 2025. 2

[23] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018. 2, 6

[24] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3

[25] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[26] Linhao Zhong, Fan Li, Yi Huang, Jianzhuang Liu, Renjing Pei, and Fenglong Song. Outdreamer: Video outpainting with a diffusion transformer. *arXiv preprint arXiv:2506.22298*, 2025. 2

[27] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *ICCV*, pages 10477–10486, 2023. 3, 6

[28] Bojia Zi, Weixuan Peng, Xianbiao Qi, Jianan Wang, Shihao Zhao, Rong Xiao, and Kam-Fai Wong. Minimax-remover: Taming bad noise helps video object removal. *NeurIPS*, 2025. 2, 3, 6, 7, 8