BEYOND DENSE CONTEXT: EFFICIENT VIDEO INPAINTING VIA POSITION-AWARE CONTEXT EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video inpainting is inherently asymmetric, as only the masked region requires synthesis, while the non-mask region primarily provides context. Existing efficient methods either summarize non-mask content into spatially coarse global features or estimate dense spatial features, creating a trade-off between computational efficiency and fine-grained context preservation. To understand which context is needed, we analyze masked-to-non-mask attention in a dense baseline. We find that masked queries concentrate their attention on a small subset of non-mask tokens, particularly those near the mask and in semantically informative regions. Motivated by this observation, we introduce **PACE**, a video inpainting framework that extracts compact, position-aware context. PACE assigns sparse spatial anchors to a small set of learnable queries. Through attention that incorporates both content and relative spatiotemporal position, these queries aggregate non-mask information from across the video into compact context features. An adaptive attention fusion mechanism then dynamically balances the extracted context with mask-side information. By compressing dense non-mask tokens into compact context representations, PACE accelerates masked-region denoising while preserving the contextual information needed for high-quality inpainting. Experiments on DAVIS, YouTube-VOS, and AdErase, our new advertising video benchmark for text and logo removal, demonstrate strong inpainting quality. PACE outperforms the efficient baselines EditCtrl and YOSE in quality and achieves an average $1.7\times$–$2.1\times$ speedup over them across different mask ratios.

1 INTRODUCTION

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have shown strong performance in video inpainting (Wan Team et al., 2025; Jiang et al., 2025; Bian et al., 2025; Zi et al., 2025a). Dense DiTs represent the video as a sequence of spatiotemporal tokens and jointly process tokens from both masked and non-mask regions. These regions, however, serve different roles in inpainting: only the masked region requires synthesis, while the non-mask region primarily provides context. This distinction is especially relevant to subtitle and logo removal, where the masked region often occupies only a small portion of the video. Processing the full token sequence therefore incurs substantial computational cost even when little content needs to be synthesized.

To reduce this cost, recent methods (Litman et al., 2026; Wu et al., 2026a) concentrate computation on masked regions while retaining contextual information from non-mask regions. As illustrated in Fig. 1, EditCtrl (Litman et al., 2026) downsamples the video and encodes the non-mask regions as global conditioning, reducing computation but potentially losing details needed for synthesis. YOSE (Wu et al., 2026a) instead reconstructs dense key–value features for non-mask regions, preserving detailed context but requiring masked queries to attend to the full non-mask token set. *However, how much non-mask context is necessary for synthesis remains unclear.*

To investigate this question, we examine attention from masked queries to non-mask tokens in the dense base model (Fig. 2). We find that attention is concentrated on a small subset of tokens. These highly attended tokens often appear near the mask, where they can support continuity across its boundary, or in regions that provide appearance and structural cues for the missing content. This

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

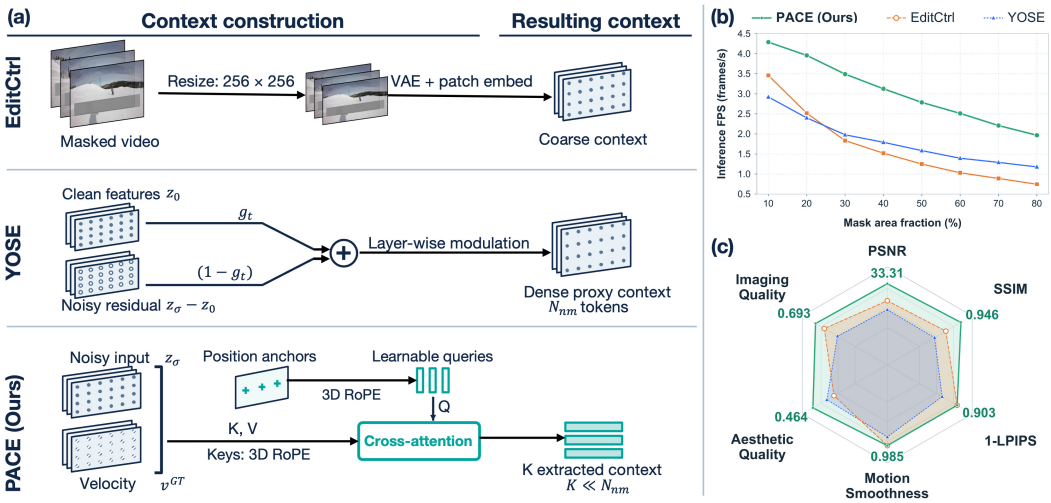


Figure 1: (a) Context construction in EditCtrl, YOSE, and PACE. (b) Inference FPS across mask ratios. (c) Quality comparison on AdErase. Axes are scaled separately, with $1 - \text{LPIPS}$ shown so that higher values are better in every panel.

uneven attention distribution motivates learning a compact context representation that gives greater weight to informative non-mask regions when aggregating background cues.

We propose **PACE** to construct compact context representations at each DiT block. The dense DiT takes the latent state $\mathbf{z}_\sigma$ as input and is trained to predict the flow velocity $\mathbf{v}^{\mathrm{GT}}$, with intermediate non-mask features providing context for the masked branch. Since both the input and the output target are known at non-mask locations, we use them jointly to learn compact approximations of these intermediate features. Specifically, we compress these two signals using a small set of learnable queries placed at sparse positions. Guided by positional information, the queries gather relevant non-mask content across the video while retaining spatial cues needed for inpainting. Adaptive attention fusion then balances this context with mask-side information. It compensates for the reduced context token count and uses a learned correction conditioned on current features to adjust the relative contributions of the two sources. The resulting compact representations reduce the cost of masked-to-non-mask attention to accelerate inference, while retaining the contextual information needed for high-quality inpainting results.

We evaluate PACE on DAVIS (Perazzi et al., 2016), YouTube-VOS (Xu et al., 2018), and AdErase, our new e-commerce advertising benchmark featuring multiple small text and logo masks per video. Across all three benchmarks, PACE improves non-mask preservation and visual quality over the efficient baselines EditCtrl and YOSE, matching or exceeding both methods on every reported metric. Compared with dense full-video methods such as VACE, it achieves stronger non-mask preservation while maintaining competitive visual quality, indicating that compact context can retain much of the information needed for high-quality synthesis. Alongside these quality benefits, PACE achieves an average $1.7\times$–$2.1\times$ speedup over EditCtrl and YOSE across the evaluated mask ratios (Fig. 1(b)). The three main contributions of this work are summarized below:

- We show that masked-to-non-mask attention in the dense model concentrates on a small subset of non-mask tokens, often near mask boundaries or in regions providing appearance and structural cues. This observation motivates compact context extraction.
- We introduce **PACE**, which learns compact non-mask features for each DiT block from the known input state and target flow velocity. Position-aware learnable queries extract these features, and adaptive attention fusion incorporates them into masked-region synthesis.
- Extensive evaluations on AdErase, DAVIS, and YouTube-VOS establish PACE's quality and efficiency advantages. PACE improves quality over EditCtrl and YOSE across all three benchmarks, remains competitive with full-video methods, and achieves faster inference across the full range of evaluated mask ratios.

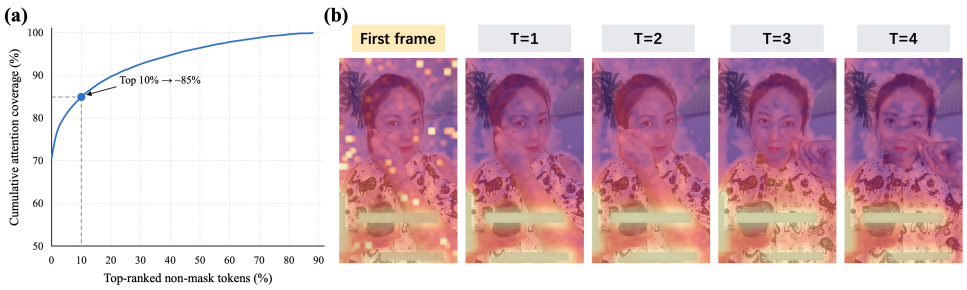


Figure 2: **(a)** With mask tokens retained, few non-mask tokens suffice for high attention coverage. **(b)** Attention maps reveal concentration near the mask and in semantically informative regions, together supporting compact, position-aware context representations.

2 Related Work

Video inpainting and removal. Early video inpainting methods recover missing content by exploiting correspondences across frames through optical flow and spatiotemporal feature aggregation (Huang et al., 2016; Gao et al., 2020; Zeng et al., 2020; Liu et al., 2021; Li et al., 2022; Zhang et al., 2022; Zhou et al., 2023). These approaches benefit from content visible in other frames, but persistent occlusions and inaccurate correspondences hinder faithful completion. Diffusion models provide generative priors for synthesizing missing content (Ho et al., 2020; Rombach et al., 2022; Lugmayr et al., 2022; Zhang et al., 2024; Zi et al., 2025b). Other methods combine generation with propagated context, including flow-guided priors (Li et al., 2025a), noise propagation (Lee et al., 2025), and homography-based propagation (Ding et al., 2025). Large video diffusion backbones such as Wan (Wan Team et al., 2025) further support generative inpainting and editing. VACE (Jiang et al., 2025) unifies video creation and editing through a common conditioning framework, while VideoPainter (Bian et al., 2025) injects background context into pretrained video DiTs through a lightweight context encoder. MiniMax-Remover (MiniMax-RM) (Zi et al., 2025a) combines a simplified removal architecture with minimax distillation to improve removal quality and sampling efficiency. ROSE (Miao et al., 2025) and EffectErase (Fu et al., 2026) address object-related effects such as shadows and reflections, while CLEAR (He et al., 2026) targets subtitle removal with mask-free inference. Recent work also allocates computation according to the edit region. EditCtrl (Litman et al., 2026) restricts denoising to masked tokens and combines local control with a lightweight global context embedder operating on a downsampled background video. YOSE (Wu et al., 2026a) selects masked tokens through variable-length indexing and estimates proxy key–value features for non-mask tokens. These methods obtain context through spatial downsampling or dense feature estimation. PACE instead learns to select and compress relevant non-mask information into a small set of position-aware context features for masked-region synthesis.

Efficient video diffusion and token reduction. Video diffusion acceleration targets either the cost of each denoising step or the number of sampling steps. Methods that lower per-step cost include sparse attention, which restricts token interactions based on spatial and temporal structure (Xi et al., 2025; Yang et al., 2025; Zhang et al., 2025b; Li et al., 2025b; Zhang et al., 2025a; Wu et al., 2026b). Recent work further improves sparse attention through finer-grained block selection, offline calibration, attention distillation, and reuse of sparsity patterns (Hu et al., 2026; Yehezkel et al., 2026; Han et al., 2026; Lin et al., 2026). Token merging offers another way to reduce attention cost by shortening the token sequence (Bolya et al., 2023; Bolya & Hoffman, 2023; Lu et al., 2025; Li et al., 2024). Feature caching instead exploits redundancy across denoising steps, reusing or predicting intermediate features to avoid repeated computation (Zhao et al., 2025; Liu et al., 2025; Kahatapitiya et al., 2025; Lv et al., 2025; Zheng et al., 2026). To reduce the number of sampling steps, consistency models and distillation methods learn to generate samples with fewer model evaluations (Song et al., 2023; Yin et al., 2024). In video generation, TMD pursues this goal by matching a multi-step denoising trajectory with a few-step transition process (Nie et al., 2026). PACE reduces per-step cost through the asymmetry of video inpainting. Only masked regions require synthesis, motivating learned compression of the non-mask information that provides context.

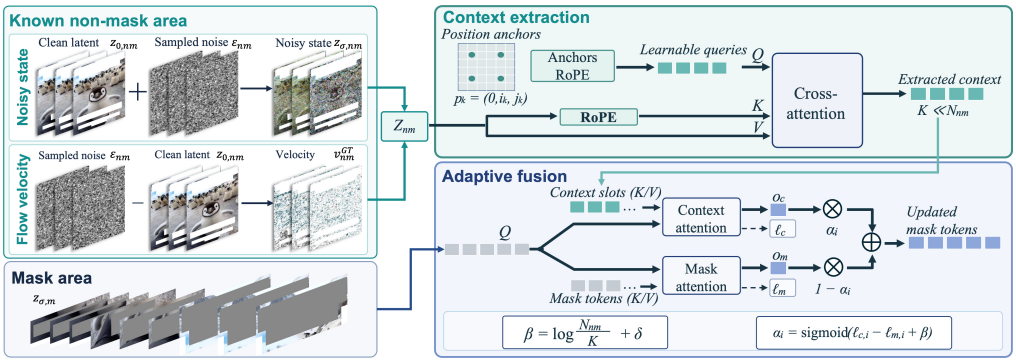


Figure 3: **Overview of PACE.** PACE extracts position-aware context slots from the known states. Masked queries attend to both mask tokens and compact context, with the two sources combined by adaptive fusion at each DiT block.

3 METHOD

Let Ω_{m} and Ω_{nm} denote the masked and non-mask regions, containing M and N_{nm} tokens, respectively. PACE propagates only the M masked tokens through the transformer blocks. At each block, $K \ll N_{\mathrm{nm}}$ context slots summarize the known non-mask trajectory, and adaptive attention fusion combines attention to these slots with mask-side attention.

3.1 POSITION-AWARE CONTEXT EXTRACTION

Known non-mask trajectory. Given the clean non-mask latent $\mathbf{z}_0$ and sampled noise $\boldsymbol{\varepsilon}$, we compute the non-mask state and flow velocity at timestep σ:

$$\mathbf{z}_\sigma\big|_{\Omega_{\mathrm{nm}}} = (1-\sigma)\mathbf{z}_0\big|_{\Omega_{\mathrm{nm}}} + \sigma\boldsymbol{\varepsilon}\big|_{\Omega_{\mathrm{nm}}}, \qquad \mathbf{v}^{\mathrm{GT}}\big|_{\Omega_{\mathrm{nm}}} = \boldsymbol{\varepsilon}\big|_{\Omega_{\mathrm{nm}}} - \mathbf{z}_0\big|_{\Omega_{\mathrm{nm}}}. \tag{1}$$

Let E_z and E_v denote their embeddings into the DiT's D-dimensional feature space. Concatenating along the token axis gives $\mathbf{Z}_{\mathrm{nm}} = [E_z(\mathbf{z}_\sigma)|_{\Omega_{\mathrm{nm}}}; E_v(\mathbf{v}^{\mathrm{GT}})|_{\Omega_{\mathrm{nm}}}] \in \mathbb{R}^{2N_{\mathrm{nm}} \times D}$. Each non-mask location contributes two tokens, one from the latent state and one from the flow velocity. Both use the spatiotemporal coordinates of that location for positional encoding. Embedding details are given in section B.1 of the supplementary material.

Sparse spatial anchors. We construct a sparse grid on the first-frame token grid of size $H_p \times W_p$. We divide its longer and shorter spatial axes into n_{long} and n_{short} intervals, respectively, forming $K = n_{\mathrm{long}} n_{\mathrm{short}}$ tiles. The center token of each tile defines an anchor:

$$\mathbf{p}_k = (0, i_k, j_k), \qquad k = 1, \ldots, K, \tag{2}$$

where 0 denotes the first frame and (i_k, j_k) are the center token's row and column coordinates in the original token grid. The positional encodings associated with these anchors will be applied to learnable queries in the following context extraction step.

Position-aware context extraction. At block l, a bank of learnable queries $\mathbf{Q}^{(l)} \in \mathbb{R}^{K \times D}$ is associated with the anchors $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{K}$. The query bank is layer-specific, while the extraction projections are shared across layers. Using row-wise layer normalization LN, we compute:

$$\mathbf{Q}_{\mathrm{c}}^{(l)} = \mathrm{RoPE}_{\mathcal{P}}\left(\mathrm{LN}_Q(\mathbf{Q}^{(l)})\mathbf{W}_Q\right), \tag{3}$$

$$\mathbf{K}_{\mathrm{c}} = \mathrm{RoPE}_{\mathcal{P}_{\mathrm{nm}}^{(2)}}\left(\mathrm{LN}_K(\mathbf{Z}_{\mathrm{nm}})\mathbf{W}_K\right), \tag{4}$$

$$\mathbf{V}_{\mathrm{c}} = \mathbf{Z}_{\mathrm{nm}}\mathbf{W}_V, \tag{5}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D \times D}$, and projection biases are omitted for clarity. The coordinates $\mathcal{P}_{\mathrm{nm}}^{(2)}$ repeat the non-mask positions for the two source fields. Multi-head attention aggregates their embedded features into K context slots as follows:

$$\mathbf{S}^{(l)} = \mathrm{Attn}\left(\mathbf{Q}_{\mathrm{c}}^{(l)}, \mathbf{K}_{\mathrm{c}}, \mathbf{V}_{\mathrm{c}}\right)\mathbf{W}_O, \qquad \mathbf{S}^{(l)} \in \mathbb{R}^{K \times D}, \tag{6}$$

where $\mathbf{W}_O \in \mathbb{R}^{D\times D}$ is the output projection. RoPE supplies relative spatial and temporal cues during extraction. The extracted slots are converted into context keys $\mathbf{K}_{\text{slot}}^{(l)}$ and values $\mathbf{V}_{\text{slot}}^{(l)}$ for fusion with the masked branch; implementation details are provided in section B.1.

3.2 Adaptive Fusion

Each masked query attends to mask tokens and context slots. Since context compression changes the key count and hence the softmax balance, we calibrate the two branches using their attention statistics and features. Omitting layer and head indices, their outputs and log-normalizers are:

$$(\mathbf{o}_{\text{m}}, \ell_{\text{m}}) = \text{Attn}(\mathbf{Q}_{\text{m}}, \mathbf{K}_{\text{m}}, \mathbf{V}_{\text{m}}), \quad (\mathbf{o}_{\text{c}}, \ell_{\text{c}}) = \text{Attn}(\mathbf{Q}_{\text{m}}, \mathbf{K}_{\text{slot}}, \mathbf{V}_{\text{slot}}), \tag{7}$$

Here $\mathbf{o}$ is the softmax-weighted value sum and ℓ the log-sum-exp of attention logits for each query; m and c denote mask and context branches (definitions in section C).

Correcting for context compression. The branch normalizers reflect both content relevance and key count. Compressing N_{nm} non-mask tokens into K slots reduces the context branch's attention mass when logit magnitudes are comparable. To compensate, we treat each slot as representing N_{nm}/K tokens and add a count correction to its logits:

$$\beta_{\text{cnt}} = \log\big(N_{\text{nm}}/K\big), \tag{8}$$

This scales the unnormalized context mass by N_{nm}/K under an equal-group-size assumption, and does not recover the dense logits.

Learning the residual correction. To account for context relevance beyond token counts, we condition a learned residual on the mean mask hidden state $\bar{\mathbf{x}}_{\text{m}} \in \mathbb{R}^D$ and mean slot value $\bar{\mathbf{v}}_{\text{slot}} \in \mathbb{R}^d$ for the current head of dimension d:

$$\delta = b + \tau \tanh\left(c + \langle \bar{\mathbf{x}}_{\text{m}}, \mathbf{w}_{\text{m}} \rangle + \langle \bar{\mathbf{v}}_{\text{slot}}, \mathbf{w}_{\text{c}} \rangle\right). \tag{9}$$

The total correction is $\beta = \beta_{\text{cnt}} + \delta$. The parameters $b, c, \mathbf{w}_{\text{m}}, \mathbf{w}_{\text{c}}$ are learned per layer and head, with $\tau = 3$. Padding tokens are excluded from the means. Parameter dimensions and initialization are given in section B.1.

Calibrated fusion. For masked query i, the context weight and fused output are:

$$\alpha_i = \text{sigmoid}\left(\ell_{\text{c},i} - \ell_{\text{m},i} + \beta\right), \tag{10}$$

$$\mathbf{o}_i = (1-\alpha_i)\mathbf{o}_{\text{m},i} + \alpha_i \mathbf{o}_{\text{c},i}. \tag{11}$$

The correction β adjusts the branch balance while preserving each branch's internal attention distribution. Setting $\beta = 0$ recovers standard online-softmax merging (section C).

3.3 Training and Inference

Training. We train PACE with a flow-matching loss over the masked tokens:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\sigma,\boldsymbol{\varepsilon}}\left[\text{MSE}\left(\hat{\mathbf{v}}_{\text{m}}, (\boldsymbol{\varepsilon} - \mathbf{z}_0)\big|_{\Omega_{\text{m}}}\right)\right], \tag{12}$$

where $\hat{\mathbf{v}}_{\text{m}}$ is the predicted masked-region velocity at timestep σ, and MSE averages over the masked tokens and the output dimensions.

Inference. At each sampling step, PACE updates the masked tokens using context extracted from the known non-mask trajectory. After sampling, we assemble the full latent grid from the generated masked tokens and the original non-mask latents:

$$\hat{\mathbf{z}}_p = \begin{cases} \hat{\mathbf{z}}_{\text{m},p}, & p \in \Omega_{\text{m}}, \\ \mathbf{z}_{0,p}, & p \in \Omega_{\text{nm}}. \end{cases} \tag{13}$$

The VAE decodes this assembled grid into the output video.

4 EXPERIMENTS

Datasets. We train on approximately 400K commercial advertising videos collected from the Internet, sampling random masks from text-free regions to construct training pairs. For evaluation, we introduce **AdErase**, a benchmark of 450 e-commerce advertising clips containing subtitles, small logos, or both. These evaluation videos do not overlap with the training set. Removal masks are rasterized from the detected overlay bounding boxes. We also evaluate on 90 sequences each from DAVIS (Perazzi et al., 2016) and YouTube-VOS (Xu et al., 2018), following the evaluation protocol used by YOSE (Wu et al., 2026a).

Training. We initialize PACE from Wan2.1-Fun-V1.1-1.3B-InP (Wan Team et al., 2025) and jointly train a rank-64 task LoRA and the PACE module for 6K steps on 49-frame 480×832 clips with an effective batch size of 64 and AdamW ($\beta_2 = 0.95$); learning rates are 10^{-4} for the LoRA and 10^{-3} for PACE. EditCtrl and YOSE are fine-tuned on the same training data for fair comparison.

Metrics. All methods are evaluated on 49-frame clips at 480×832 resolution. We measure *non-mask preservation* using PSNR, SSIM, and LPIPS (Zhang et al., 2018) outside the removal mask, with the input video as reference. We assess *overall video quality* using Motion Smoothness, Aesthetic Quality, and Imaging Quality (Huang et al., 2024). All metrics are computed on model outputs without post-hoc pixel-space replacement of the non-mask region. These metric groups assess preservation of known content and the perceptual quality of the complete video, respectively.

4.1 FLOPS AND INFERENCE EFFICIENCY

We first analyze the computation of attention scores and value aggregation. Let N and M denote the numbers of full-video and selected mask-region tokens, respectively, with $N_{\mathrm{nm}} = N - M$, K context slots, and feature dimension D. These operations require $\mathcal{O}(N^2 D)$ computation in dense self-attention. For PACE, the corresponding complexity is:

$$G_{\mathrm{attn}}^{\mathrm{PACE}} = \mathcal{O}\left(M^2 D + MKD + 2KN_{\mathrm{nm}}D\right), \tag{14}$$

where the three terms correspond to mask-side attention, mask-to-slot attention, and context extraction. The coefficient 2 in $2KN_{\mathrm{nm}}D$ accounts for the N_{nm} state tokens and N_{nm} velocity tokens used for context extraction. With the same feature dimension, the estimated reduction factor for these attention operations is:

$$R_{\mathrm{attn}} \approx \frac{N^2}{M^2 + MK + 2KN_{\mathrm{nm}}}. \tag{15}$$

For the $13 \times 30 \times 52$ token grid, $N = 20{,}280$. With $K = 91$ and $M/N = 20\%$, we obtain $R_{\mathrm{attn}} \approx 20.8$. This estimate covers only attention-score computation and value aggregation, excluding projections, feed-forward layers, and other model operations; it is not an end-to-end FLOPs or runtime reduction factor for the complete model.

We evaluate practical inference efficiency on a single A100 GPU using 49-frame clips at 480×832 resolution and 25 denoising steps for PACE, EditCtrl, and YOSE (protocol in section E). As shown in Fig. 1(b), PACE achieves higher inference FPS than EditCtrl and YOSE at all evaluated mask ratios from 10% to 80%. Averaging the speedup across these mask ratios for each baseline yields $1.7\times$–$2.1\times$ across the two comparisons. This range summarizes the baseline-specific averages, rather than the minimum and maximum speedups at individual mask ratios. PACE's FPS decreases as the mask ratio increases, consistent with the growing cost of processing more masked tokens. The relative speedup also depends on each baseline's computation and implementation, so attention complexity alone does not determine the observed trend.

4.2 QUALITATIVE RESULTS

CLEAR takes the original videos as input because it performs mask-free removal. For PACE and the remaining baselines, we fill the detected bounding boxes with gray and ask each method to complete these regions. This provides the same input videos and the same completion task for the mask-based methods, ensuring a consistent comparison.

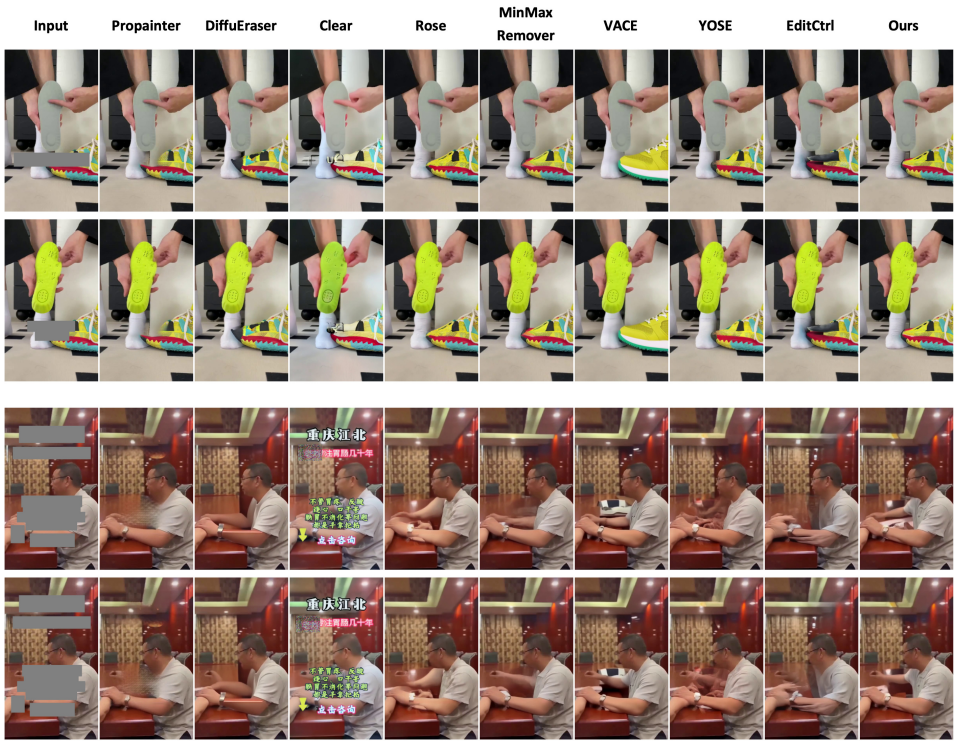


Figure 4: Visual comparison on AdErase. PACE achieves better inpainting quality than ProPainter, DiffuEraser, and efficient baselines (YOSE, EditCtrl) in the examples shown, while producing results comparable to full-video generative methods (ROSE, MiniMax-Remover, VACE). The input column shows the original videos with removal boxes overlaid.

Fig. 4 compares the visual quality of the completed regions on AdErase. ProPainter produces blurry completions, while DiffuEraser also leaves relatively soft details despite incorporating diffusion priors. CLEAR leaves visible overlay remnants in some examples. Compared with the efficient baselines YOSE and EditCtrl, PACE produces more detailed content that blends more naturally with the surrounding regions. ROSE, VACE, and MiniMax-Remover process the full video and produce detailed completions. Despite using compact non-mask context, PACE achieves visual quality comparable to these full-video methods in the examples shown. This suggests that its lightweight context representation preserves the information needed to reconstruct local details and maintain consistency with the surrounding visible background.

4.3 Quantitative Comparison

Advertising benchmark (AdErase). As shown in Tab. 1, PACE ranks first or second on all six metrics among the compared methods. Against the efficient baselines EditCtrl and YOSE, it improves PSNR by 0.70 and 1.07 dB, respectively, and achieves higher SSIM, Aesthetic Quality, and Imaging Quality. Its LPIPS and Motion Smoothness match EditCtrl and improve on YOSE. These results show gains in both non-mask preservation and visual quality with compact context.

Compared with the full-video baseline VACE, PACE achieves higher PSNR (33.31 versus 32.26 dB) and SSIM (0.946 versus 0.943), while matching LPIPS (0.097) and Motion Smoothness (0.985). It also achieves higher Aesthetic Quality (0.464 versus 0.459) and nearly identical Imaging Quality (0.693 versus 0.694). Thus, despite compressing non-mask context, PACE maintains visual quality comparable to this full-video method while improving preservation of the known regions.

CLEAR ties for the highest Motion Smoothness (0.986), with PSNR of 22.29 dB and Imaging Quality of 0.677. Its input differs from that of the mask-based methods, as it performs mask-free removal on the original videos rather than completing predefined grayed-out regions. Processing the

Table 1: Quantitative comparison on AdErase. Bold and underline mark the best and second-best distinct scores, including ties (as in Tab. 2). PACE ranks first or second on all six metrics.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Mot. Smo. ↑	Aes. Qua. ↑	Ima. Qua. ↑
ProPainter	31.63	0.923	0.148	0.983	0.459	0.688
DiffuEraser	31.51	0.918	0.113	0.982	0.456	0.686
MiniMax-RM	32.18	0.942	**0.097**	0.984	0.461	<u>0.693</u>
ROSE	31.43	0.935	0.106	**0.986**	<u>0.462</u>	0.687
VACE	32.26	<u>0.943</u>	**0.097**	<u>0.985</u>	0.459	**0.694**
CLEAR	22.29	0.831	0.182	**0.986**	0.458	0.677
EditCtrl	<u>32.61</u>	0.942	**0.097**	<u>0.985</u>	0.458	0.691
YOSE	32.24	0.939	<u>0.099</u>	0.984	0.460	0.688
PACE (Ours)	**33.31**	**0.946**	**0.097**	<u>0.985</u>	**0.464**	<u>0.693</u>

Table 2: Quantitative comparison on DAVIS and YouTube-VOS. PACE achieves stronger non-mask preservation than the full-video method VACE while approaching its visual quality.

Method	**DAVIS**						**YouTube-VOS**					
	PSNR ↑	SSIM ↑	LPIPS ↓	Mot. Smo. ↑	Aes. Qua. ↑	Ima. Qua. ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Mot. Smo. ↑	Aes. Qua. ↑	Ima. Qua. ↑
ProPainter	**32.48**	**0.863**	**0.088**	0.965	0.416	0.563	32.84	**0.908**	0.074	0.972	0.374	0.532
DiffuEraser	31.19	0.782	0.139	**0.969**	0.427	0.553	32.33	0.869	0.090	0.973	0.392	0.541
MiniMax-RM	31.23	0.813	0.125	<u>0.968</u>	0.434	0.573	32.72	0.904	<u>0.072</u>	<u>0.975</u>	0.396	0.546
ROSE	29.63	0.787	0.143	**0.969**	0.438	0.579	30.09	0.883	0.095	**0.976**	0.398	0.549
VACE	31.60	0.811	0.126	0.967	**0.460**	**0.607**	<u>33.43</u>	0.904	0.073	0.972	**0.418**	**0.581**
EditCtrl	29.52	0.799	0.151	0.962	0.440	0.562	31.31	0.894	0.092	0.968	0.391	0.547
YOSE	30.05	<u>0.819</u>	0.133	0.967	0.428	0.579	32.77	0.901	0.077	0.974	0.392	0.540
PACE (Ours)	<u>31.77</u>	<u>0.819</u>	<u>0.123</u>	<u>0.968</u>	<u>0.441</u>	<u>0.589</u>	**33.63**	<u>0.907</u>	**0.069**	0.974	<u>0.405</u>	<u>0.570</u>

full frame can also alter background content outside the removal regions, which may contribute to its lower non-mask preservation scores on this benchmark.

Public benchmarks (DAVIS & YouTube-VOS). Tab. 2 reports results on both public benchmarks. Against the efficient baselines EditCtrl and YOSE, PACE matches or exceeds every reported metric on both datasets. Its PSNR gains over EditCtrl are 2.25 dB on DAVIS and 2.32 dB on YouTube-VOS, and the corresponding gains over YOSE are 1.72 and 0.86 dB. Higher Aesthetic Quality and Imaging Quality scores on both datasets show that these preservation gains are also accompanied by improved visual quality.

The propagation-based method ProPainter achieves higher PSNR and SSIM and lower LPIPS than PACE on DAVIS. Its strength in preserving observed content is consistent with its use of optical-flow-guided propagation. However, these metrics are evaluated outside the mask and do not directly measure the quality of the completed regions. PACE scores higher than ProPainter in Motion Smoothness, Aesthetic Quality, and Imaging Quality on both public benchmarks. The qualitative examples on AdErase in Fig. 4 also show sharper details and more natural completions from PACE, whereas ProPainter produces blurrier content. These observations support PACE's perceptual advantages in video completion despite ProPainter's stronger non-mask preservation on DAVIS.

Against the full-video generative baseline VACE, PACE achieves stronger non-mask preservation despite using compact context, with higher PSNR and SSIM and lower LPIPS on both datasets. Its Motion Smoothness is also slightly higher. VACE retains the best Aesthetic Quality and Imaging Quality, with PACE ranking second in both metrics on both datasets. This comparison places PACE close to the full-video baseline in visual quality while preserving the known background more accurately across both of the public evaluation datasets. Together, these results suggest that dense attention over the full video is not necessary for strong completion quality.

Table 3: Component ablation on the YouTube-VOS benchmark.

Metric	w/o context	w/o calibration	PACE (Ours)
PSNR↑	32.26	33.32	33.63
SSIM↑	0.901	0.906	0.907
LPIPS↓	0.080	0.073	0.069
Mot. Smo.↑	0.973	0.974	0.974
Aes. Qua.↑	0.394	0.396	0.405
Ima. Qua.↑	0.546	0.555	0.570

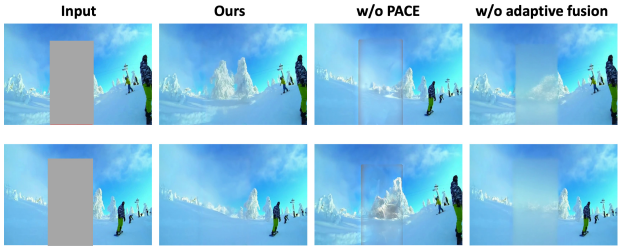


Figure 5: Qualitative ablation of context extraction and attention calibration for the video inpainting task.

4.4 Analysis

Component ablation. Tab. 3 examines context extraction and attention calibration on YouTube-VOS. Removing the context branch under the same backbone adaptation lowers Imaging Quality by 0.024 and non-mask PSNR by 1.37 dB. Masked queries then lack background context, which may reduce consistency with their surroundings. This ablation tests the complete branch, including its positional information. Although non-mask latents remain unchanged, joint VAE decoding with generated mask latents can affect non-mask pixels near the edited boundary.

The *w/o calibration* variant retains both attention branches but merges their outputs using standard online-softmax ($\beta = 0$). Restoring attention calibration improves PSNR from 33.32 to 33.63 dB, reduces LPIPS from 0.073 to 0.069, and raises Aesthetic Quality and Imaging Quality by 0.009 and 0.015, respectively, with unchanged Motion Smoothness. Calibrating the slots' contribution thus improves both preservation and visual quality. At the same slot budget, these gains show that calibrating the context contribution matters beyond retaining the context branch. This ablation evaluates the count term and learned residual jointly. In Fig. 5, "w/o PACE" and "w/o adaptive fusion" denote context removal and uncalibrated fusion, respectively.

Slot budget K. Tab. 4 examines the quality–cost trade-off on YouTube-VOS. Increasing K from 18 to 91 improves PSNR by 0.20 dB, reduces LPIPS from 0.073 to 0.069, and raises Imaging Quality from 0.556 to 0.570. Increasing K to 390 adds another 0.19 dB in PSNR and slightly improves Motion Smoothness, but leaves SSIM unchanged and marginally lowers the other perceptual scores. It also increases the slot-dependent attention operations by approximately $4.3\times$ relative to $K = 91$ (Eq. (14)), without implying the same increase in end-to-end runtime. We therefore use $K = 91$ as a practical balance between quality and inference cost.

Table 4: Context-slot budget on YouTube-VOS. $K = 91$ balances quality and cost.

Anchor grid (K)	PSNR ↑	SSIM ↑	LPIPS ↓	Mot. Smo. ↑	Aes. Qua. ↑	Ima. Qua. ↑
26×15 ($K = 390$)	33.82	0.907	0.070	0.975	0.403	0.567
13×7 ($K = 91$, **Ours**)	33.63	0.907	0.069	0.974	0.405	0.570
6×3 ($K = 18$)	33.43	0.906	0.073	0.974	0.397	0.556

5 Conclusion

We presented **PACE**, an efficient video inpainting framework motivated by the distinct roles of masked regions and non-mask context. Position-aware learnable queries compress the known input state and target flow velocity into compact context, while adaptive fusion controls its contribution to masked-region synthesis. Together, these components retain relevant background cues while reducing masked-to-non-mask attention cost. On DAVIS, YouTube-VOS, and AdErase, PACE improves non-mask preservation and visual quality over EditCtrl and YOSE and remains competitive with full-video generative methods. Its average $1.7\times$–$2.1\times$ speedup over the two efficient baselines across mask ratios demonstrates the practical benefit of learned context compression.

AI USE STATEMENT

We used generative AI tools for language polishing, manuscript consistency checks, and assistance with the presentation and interpretation of results. We also used these tools to assist with literature searches and deeper exploration of related work, starting from relevant papers and tracing both their references and subsequent works that cite them. This process helped us identify related research directions and organize the connections and progression of prior work. The authors take full responsibility for the final content and conclusions of this paper.

REPRODUCIBILITY STATEMENT

To support reproducibility, section 3 specifies the context extraction and adaptive fusion mechanisms, training objective, and inference procedure. Section 4 reports the pretrained backbone, data sources, training hyperparameters, evaluation metrics, and hardware and denoising-step settings for the efficiency comparison. The evaluation protocol specifies clip length and resolution and measures model outputs without post-hoc replacement of non-mask pixels. Section 4.4 reports component and context-slot ablations, while sections A and C to E provide the attention measurement protocol, fusion derivation, workload mask-ratio statistics, and timing protocol underlying the analysis. We will publicly release the code and evaluation data upon acceptance.

REFERENCES

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*, 2025.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *CVPRW*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023.

Ding Ding, Yueming Pan, Ruoyu Feng, Qi Dai, Kai Qiu, Jianmin Bao, Chong Luo, and Zhenzhong Chen. HomoGen: Enhanced video inpainting via homography propagation and diffusion. In *CVPR*, 2025.

Yang Fu, Yike Zheng, Ziyun Dai, and Henghui Ding. EffectErase: Joint video object removal and insertion for high-quality effect erasing. In *CVPR*, 2026.

Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020.

Shihao Han, Hao Yang, Xinting Hu, Xiaofeng Mei, Yi Jiang, and Xiaojuan Qi. Veda: Scalable video diffusion via distilled sparse attention. In *ICML*, 2026.

Qingdong He, Chaoyi Wang, Peng Tang, Yifan Yang, and Xiaobin Hu. CLEAR: Context-aware learning with end-to-end mask-free inference for adaptive video subtitle removal. In *ICML*, 2026.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jie Hu, Zixiang Gao, Yutong He, and Kun Yuan. DFSAttn: Dynamic fine-grained sparse attention for efficient video generation. In *ICML*, 2026.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *TOG*, 2016.

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*, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *ICCV*, 2025.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S. Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *ICCV*, 2025.

Minhyeok Lee, Suhwan Cho, Chajin Shin, Jungho Lee, Sunghun Yang, and Sangyoun Lee. Video diffusion models are strong video inpainter. In *AAAI*, 2025.

Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. DiffuEraser: A diffusion model for video inpainting. *arXiv:2501.10018*, 2025a.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial Attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. In *NeurIPS*, 2025b.

Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. VidToMe: Video token merging for zero-shot video editing. In *CVPR*, 2024.

Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, 2022.

Shangran Lin, Lu Lu, Jian Chen, and Qiang Liu. RAPID: Reusing attention sparsity with inter-step adaptation for efficient video diffusion. In *CVPR*, 2026.

Yehonathan Litman, Shikun Liu, Dario Seyb, Nicholas Milef, Yang Zhou, Carl Marshall, Shubham Tulsiani, and Caleb Leak. EditCtrl: Disentangled local and global control for real-time generative video editing. In *CVPR*, 2026.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *CVPR*, 2025.

Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. FuseFormer: Fusing fine-grained information in transformers for video inpainting. In *ICCV*, 2021.

Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. ToMA: Token merge with attention for diffusion models. In *ICML*, 2025.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. FasterCache: Training-free video diffusion model acceleration with high quality. In *ICLR*, 2025.

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. In *NeurIPS*, 2025.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation. In *CVPR*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

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 *CVPR*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Wan Team et al. Wan: Open and advanced large-scale video generative models. *arXiv:2503.20314*, 2025.

Chenyang Wu, Lina Lei, Fan Li, Chunle Guo, Dehong Kong, Xinran Qin, Zhixin Wang, Mingming Cheng, and Chongyi Li. YOSE: You only select essential tokens for efficient DiT-based video object removal. In *CVPR*, 2026a.

Jianzong Wu, Liang Hou, Haotian Yang, Xin Tao, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. VMoBA: Mixture-of-block attention for video diffusion models. In *ICLR*, 2026b.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *ICML*, 2025.

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. *arXiv:1809.03327*, 2018.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *NeurIPS*, 2025.

Shai Yehezkel, Shahar Yadin, Noam Elata, Yaron Ostrovsky-Berman, and Bahjat Kawar. Accelerating text-to-video generation with calibrated sparse attention. In *ECCV*, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020.

Kaidong Zhang, Jingjing Fu, and Dong Liu. Flow-guided transformer for video inpainting. In *ECCV*, 2022.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *NeurIPS*, 2025a.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *ICML*, 2025b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. AVID: Any-length video inpainting with diffusion model. In *CVPR*, 2024.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *ICLR*, 2025.

Shikang Zheng, Guantao Chen, Qinming Zhou, Yuqi Lin, Lixuan He, Chang Zou, Peiliang Cai, Jiacheng Liu, and Linfeng Zhang. Let features decide their own solvers: Hybrid feature caching for diffusion transformers. In *ICLR*, 2026.

Shangchen Zhou, Chongyi Li, Kelvin C. K. Chan, and Chen Change Loy. ProPainter: Improving propagation and transformer for video inpainting. In *ICCV*, 2023.

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. In *NeurIPS*, 2025a.

Bojia Zi, Shihao Zhao, Xianbiao Qi, Jianan Wang, Yukai Shi, Qianyu Chen, Bin Liang, Rong Xiao, Kam-Fai Wong, and Lei Zhang. CoCoCo: Improving text-guided video inpainting for better consistency, controllability and compatibility. In *AAAI*, 2025b.

Supplementary Material

This supplement provides the evidence and derivations needed to assess the design choices in the main paper. The first four sections explain why non-mask context can be compressed, why the compressor uses spatial anchors and a model-aligned state–flow representation, why the branch fusion is mathematically well-defined, and why mask-local computation matches the workload distribution observed in our advertising training-data collection. The fifth section details the inference timing protocol. The final two sections provide additional qualitative comparisons of scene completion and text and logo removal.

A Attention Analysis Supporting Compact Context

This analysis examines whether masked queries attend equally to all non-mask tokens, testing the premise behind compact context extraction. Measurements on the unmodified dense backbone confirm that their attention demand is concentrated and spatially structured. This supports replacing the dense non-mask sequence with a small set of compact position-aware summaries rather than discarding this non-mask context information altogether.

Measurement protocol. We use Wan2.1-Fun-V1.1-1.3B-InP before removing any non-mask tokens. For each clip, we run the standard 25-step Flow-Euler schedule (shift 5, without classifier-free guidance) and record self-attention in all 30 transformer blocks. We retain only queries inside the erase mask, average attention over heads, and then average over blocks, denoising steps, and at most 1024 randomly sampled masked queries using a fixed seed. The resulting demand map assigns each token the attention mass it receives from the masked region and sums to one over the clip.

Concentration of non-mask demand. To obtain the oracle curve in Fig. 2(a), we rank non-mask tokens by their measured demand and accumulate them from highest to lowest, while retaining all mask tokens. Coverage includes the attention mass of both the retained mask tokens and the selected non-mask tokens. We average curves from 50 held-out DAVIS clips after interpolation to a common budget grid. At each budget, this ranking maximizes coverage for a fixed token subset under the measured average demand. It is not an upper bound on methods that select different subsets for individual queries, layers, or steps. Nor does attention coverage directly measure information preserved by a learned compressor. The concentration instead motivates compact representations whose effectiveness is assessed through inpainting results.

Spatial structure and anchor placement. The visualizations in Fig. 2(b) show concentrated responses near the mask and in semantically informative regions. PACE uses the first-frame grid as a common spatial reference for its queries, consistent with the backbone's first-frame conditioning. Each query can nevertheless retrieve non-mask content from any frame. Anchor location specifies a positional reference, rather than restricting retrieval to that frame or region.

B Design Rationale for the Context Compressor

PACE combines spatial anchors with a representation based on the DiT's known input state and target flow velocity. The anchors provide explicit spatial references, while the two source fields expose signals associated with the model's input and prediction target. The analysis below explains these choices without assuming that intermediate features can be recovered exactly from such a compact representation of the non-mask context.

Spatial references for learnable queries. Distinct learnable queries can specialize without positional anchors, but their identities alone do not specify locations in the source video. Assigning an anchor to each query introduces an explicit spatial reference through 3D RoPE. This design encourages position-aware retrieval without forcing a slot to summarize only its local neighborhood or guaranteeing that different slots contain independent information.

B.1 Embedding and Attention Details

The input-state embedding E_z uses the backbone's patch embedding with its usual conditioning channels. The velocity embedding E_v applies the same patch embedding to $\mathbf{v}^{\mathrm{GT}}$, with the conditioning channels set to zero. Both fields are gathered at non-mask token positions and concatenated along the sequence axis. No masked-region target is supplied to the context extractor at inference.

The query bank is parameterized as a shared learned bank plus a layer-specific residual. The extraction layer normalizations and Q/K/V projections are shared across layers. For each head of dimension d, the attention operation is $\mathrm{softmax}(QK^{\mathsf{T}}/\sqrt{d})V$. Head outputs are concatenated before the output projection $\mathbf{W}_O$, implemented as two linear maps with a rank-64 bottleneck. Projection biases are included in the implementation but suppressed in Eqs. (5) and (6).

Let $A_l(X) = \mathrm{LN}_l(X) \odot (1 + s_l) + t_l$ denote the backbone block's normalization and timestep-dependent scale and shift, applied identically to mask features and slots. Context keys and values are obtained using the block's existing key and value projections, with the backbone's key normalization included in the key path.

In Eq. (9), $\bar{\mathbf{x}}_{\mathrm{m}}$ averages the valid mask features after A_l and before Q/K/V projection. The context statistic $\bar{\mathbf{v}}_{\mathrm{slot}}$ averages the K projected slot values within the current head. The learned bias b, offset c, and weights $\mathbf{w}_{\mathrm{m}} \in \mathbb{R}^D$ and $\mathbf{w}_{\mathrm{c}} \in \mathbb{R}^d$ are initialized to zero, and $\tau = 3$. Thus the initial correction contains only the analytic count term. The formulas assume nonempty mask and non-mask sets. Padding positions are excluded from attention, token counts, and pooled statistics.

B.2 Affine Mixing as a Design Diagnostic

Consider a vector-valued target feature $h_l^\star$ at block l, an embedded noisy state h_{in}, and an embedded velocity h_v. A simple diagnostic approximates the target using a scalar coefficient $\eta \in \mathbb{R}$:

$$\widetilde{h}_l(\eta) = \eta h_{\mathrm{in}} + (1 - \eta)\Psi_l(h_v) + a_l, \tag{16}$$

where Ψ_l aligns the velocity embedding with the block feature space and a_l is a layer-specific offset. This unconstrained coefficient is distinct from the bounded branch-fusion weight α_i in the main text. Define $\Delta_l = h_{\mathrm{in}} - \Psi_l(h_v)$.

Proposition 1 (Optimal scalar endpoint mixing)**.** *If $\Delta_l \neq 0$, the least-squares coefficient over $\eta \in \mathbb{R}$ and its residual decomposition are given by:*

$$\eta_l^\star = \frac{\langle h_l^\star - \Psi_l(h_v) - a_l, \Delta_l \rangle}{\|\Delta_l\|^2}, \tag{17}$$

$$\|\widetilde{h}_l(\eta) - h_l^\star\|^2 = \|\widetilde{h}_l(\eta_l^\star) - h_l^\star\|^2 + (\eta - \eta_l^\star)^2 \|\Delta_l\|^2. \tag{18}$$

Proof. Writing $r = h_l^\star - \Psi_l(h_v) - a_l$ reduces the objective to $\|r - \eta\Delta_l\|^2$. Differentiation gives Eq. (17). Substituting $\eta = \eta_l^\star + u$ and using $\langle r - \eta_l^\star \Delta_l, \Delta_l \rangle = 0$ gives Eq. (18). □

Within this diagnostic model, using a fixed coefficient incurs excess error whenever it differs from the optimum for the current features. The optimum can vary with the sample, denoising state, position, and layer. This motivates learning how to combine the sources. PACE uses attention weights that vary across slots and heads, rather than one scalar coefficient. The diagnostic does not bound the reconstruction error of the deployed compressor or prove that the two sources are sufficient for exact recovery of intermediate features.

B.3 Endpoint Choice and Context Structure

PACE and YOSE (Wu et al., 2026a) both avoid propagating the full non-mask sequence through every DiT block. YOSE forms dense proxy context from clean and noisy background representations, whereas PACE compresses the known input state and target velocity into position-aware slots. To compare the source parameterizations, consider the same idealized known-background flow-matching trajectory:

$$\mathbf{z}_\sigma = (1 - \sigma)\mathbf{z}_0 + \sigma\boldsymbol{\varepsilon}, \qquad \mathbf{v}^{\mathrm{GT}} = \boldsymbol{\varepsilon} - \mathbf{z}_0, \qquad \mathbf{z}_\sigma - \mathbf{z}_0 = \sigma\mathbf{v}^{\mathrm{GT}}. \tag{19}$$

Under this shared-trajectory assumption, a displacement from the clean background contains the velocity scaled by σ. PACE instead supplies $\mathbf{z}_\sigma$ and $\mathbf{v}^{\mathrm{GT}}$ directly, corresponding to the DiT input and its training target. This comparison concerns the source parameterization and does not assume that different methods' preprocessing and VAE encodings produce identical background latents.

For $\sigma > 0$, the pairs $(\mathbf{z}_0, \mathbf{z}_\sigma)$ and $(\mathbf{z}_\sigma, \mathbf{v}^{\mathrm{GT}})$ are mutually recoverable. PACE therefore does not introduce additional information through this change of variables. It presents the velocity without the timestep-dependent scale factor, allowing the compressor to learn its contribution directly. The displacement vanishes as $\sigma \to 0$, but this observation alone does not establish a quality advantage over a learned dense estimator.

The main structural distinction is how the context is represented and read. YOSE estimates dense non-mask keys and values, while PACE learns a set of K position-aware summaries. These slots reduce the context sequence attended to by masked queries. The experiments evaluate the resulting quality–efficiency balance of the complete method.

C Exact Form of Adaptive Branch Fusion

The purpose of this derivation is to show that PACE's fusion is an ordinary softmax over the union of mask keys and context slots, with one shared bias on the context logits. Consequently, the correction changes only the total mass assigned to the context branch while preserving the relative attention distribution among its slots.

For a query i and a key subset $\mathcal{A}$, let $s_{ij} = \mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d}$ denote the attention logit for key j. Define:

$$Z_i^{\mathcal{A}} = \sum_{j \in \mathcal{A}} e^{s_{ij}}, \qquad \ell_i^{\mathcal{A}} = \log Z_i^{\mathcal{A}}, \qquad \mathbf{o}_i^{\mathcal{A}} = \frac{1}{Z_i^{\mathcal{A}}} \sum_{j \in \mathcal{A}} e^{s_{ij}} \mathbf{v}_j. \tag{20}$$

These are the output and log-sum-exp statistic used by our fused attention implementation. For two disjoint subsets $\mathcal{A}$ and $\mathcal{B}$, their exact online-softmax merge is:

$$a_i = \mathrm{sigmoid}\big(\ell_i^{\mathcal{B}} - \ell_i^{\mathcal{A}}\big), \tag{21}$$

$$\mathbf{o}_i^{\mathcal{A}\cup\mathcal{B}} = (1 - a_i)\mathbf{o}_i^{\mathcal{A}} + a_i \mathbf{o}_i^{\mathcal{B}}, \tag{22}$$

$$\ell_i^{\mathcal{A}\cup\mathcal{B}} = \log\left(e^{\ell_i^{\mathcal{A}}} + e^{\ell_i^{\mathcal{B}}}\right). \tag{23}$$

The result follows directly by substituting $e^{\ell_i^{\mathcal{A}}} \mathbf{o}_i^{\mathcal{A}}$ and $e^{\ell_i^{\mathcal{B}}} \mathbf{o}_i^{\mathcal{B}}$ for the two unnormalized value sums.

Proposition 2 (Context bias equivalence)**.** *Replacing $\ell_i^{\mathcal{B}}$ by $\ell_i^{\mathcal{B}} + \beta$ in the merge gives:*

$$\mathbf{o}_i = \frac{e^{\ell_i^{\mathcal{A}}} \mathbf{o}_i^{\mathcal{A}} + e^{\ell_i^{\mathcal{B}} + \beta} \mathbf{o}_i^{\mathcal{B}}}{e^{\ell_i^{\mathcal{A}}} + e^{\ell_i^{\mathcal{B}} + \beta}}, \tag{24}$$

which is exactly the attention obtained by adding β to every logit in $\mathcal{B}$. The within-branch distributions remain unchanged, and $\beta = 0$ recovers the unbiased online-softmax merge.

Proof. Adding the same β multiplies every unnormalized weight in $\mathcal{B}$ by e^β. This factor cancels inside $\mathbf{o}_i^{\mathcal{B}}$ but remains in the branch normalizer, yielding Eq. (24). □

For PACE, $\mathcal{A}$ contains the mask keys and $\mathcal{B}$ contains the K context slots. The correction takes the following form:

$$\beta = \log\left(\frac{N_{\mathrm{nm}}}{K}\right) + \delta \tag{25}$$

Substituting this correction recovers Eq. (10)–Eq. (11). The first term restores the key-count component lost when N_{nm} non-mask tokens are represented by K slots under the equal cluster-size interpretation. The residual δ combines a learned bias with a bounded content-dependent adjustment. Placing the correction on the normalizer adjusts the mask-versus-context mass without altering the relative attention weights within either branch.

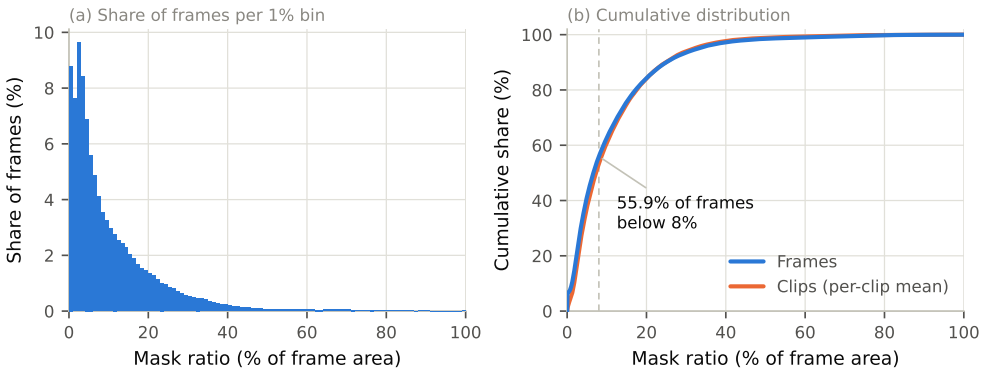


Figure 6: Removal-mask sizes in the advertising training-data collection. The left panel reports frame frequency in 1%-wide bins. The right panel reports cumulative distributions at the frame and clip levels.

Numerical implementation. We evaluate Eq. (21)–Eq. (23) using sigmoid/logsigmoid updates with fp32 accumulators. The count-normalization term is computed from each sample's actual non-mask token count. The total correction β, including the count term and learned residual, is added to the context log-sum-exp before the merge. This form exponentiates only differences between normalizers and thereby avoids the overflow risk that would arise from explicitly forming the factor e^{ℓ}.

D Mask-Ratio Statistics of Advertising Videos

We characterize removal-mask sizes using advertising videos from the training-data collection of approximately 400K videos. These workload statistics describe the source advertising data, rather than the separate 450-clip AdErase evaluation set. The training and evaluation videos do not overlap. The distribution in Fig. 6 is concentrated at small mask ratios, motivating efficient processing of small, localized removal regions.

Measurement protocol. For each analyzed frame, we construct the removal mask from the union of all active subtitle and logo boxes. We scale each box from its detection resolution to the native frame resolution and clip it to the frame boundary. Mask areas are measured before the model-side 8-pixel dilation, using the rectangle union without rasterization. A clip-level ratio is the temporal mean of its frame-level ratios. These detected overlay masks characterize removal workloads and are distinct from the random masks sampled in text-free regions to construct training pairs.

As annotated in Fig. 6, 55.9% of analyzed frames have mask ratios below 8%. Both distributions rise rapidly at small mask ratios and have a tail of larger removal regions. These observations motivate retaining masked tokens for synthesis while compressing the surrounding context. For fixed video length and slot budget, smaller masks reduce mask-side attention cost, while context extraction remains dependent on the number of non-mask tokens. The distribution therefore motivates the workload of interest, without establishing a dataset-wide speedup from mask ratios alone.

E Inference Timing Protocol

Fig. 1(b) reports inference FPS for PACE, EditCtrl, and YOSE on a single NVIDIA A100 GPU, with 25 denoising steps for each method. We use five DAVIS validation sequences (blackswan, bmx-trees, dogs-jump, gold-fish, and parkour), each represented by 49 frames at 480×832 resolution. To isolate the effect of mask area, we apply synthetic centered rectangular masks at nine area fractions spanning 5%–80%; the plot reports only the 10%–80% range. We apply a one-pixel spatial dilation (3×3) without temporal dilation, matching YOSE's selection halo. All inputs are rendered once and shared byte-identically across methods.

Timing covers the in-process pipeline call, including preparation, host-to-device transfer, and the full pipeline wall time, but excludes ffmpeg and disk I/O. At each mask fraction, we discard one warmup run and perform five timed runs. Every method processes the same five video–mask pairs. FPS is the number of frames divided by the measured pipeline wall time and is averaged over the timed repetitions. This controlled sweep keeps mask geometry consistent across methods, allowing the timing trend to be assessed as mask area increases.

F Additional Qualitative Comparisons of Scene Completion

Fig. 7 compares cases from DAVIS and YouTube-VOS. In the rock and foliage examples, ProPainter and DiffuEraser produce blurred completions, while YOSE and EditCtrl show indistinct textures or patches that do not blend with the surroundings. PACE produces sharper textures and more natural transitions to the visible content. Its results are comparable in visual detail to full-video methods such as ROSE and MiniMax-Remover. Overall, these examples show better completion quality than ProPainter, DiffuEraser, and the lightweight baselines, while remaining close to full-video methods despite using compact context.

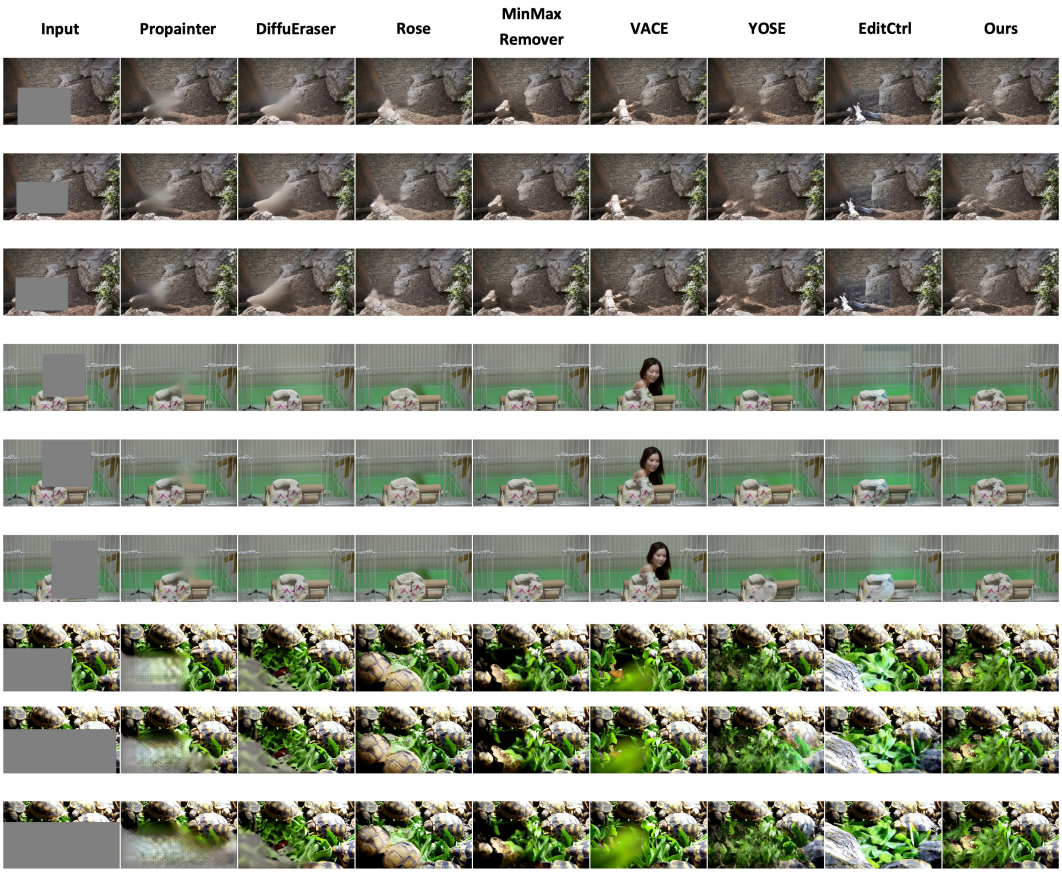


Figure 7: Additional qualitative comparisons on cases from DAVIS and YouTube-VOS. Each group contains three frames from the same video. Columns show the masked input and the compared methods; PACE is shown in the rightmost column.

G Additional Text and Logo Removal Comparisons

Fig. 8 compares removal over scene content. ProPainter leave blurred details in the tabletop examples, while CLEAR retains visible text and logos. PACE produces clearer tabletop textures and naturally completes the clothing region. Its results show better local detail than the lightweight baselines YOSE and EditCtrl and are close in visual quality to full-video methods such as ROSE and MiniMax-Remover.

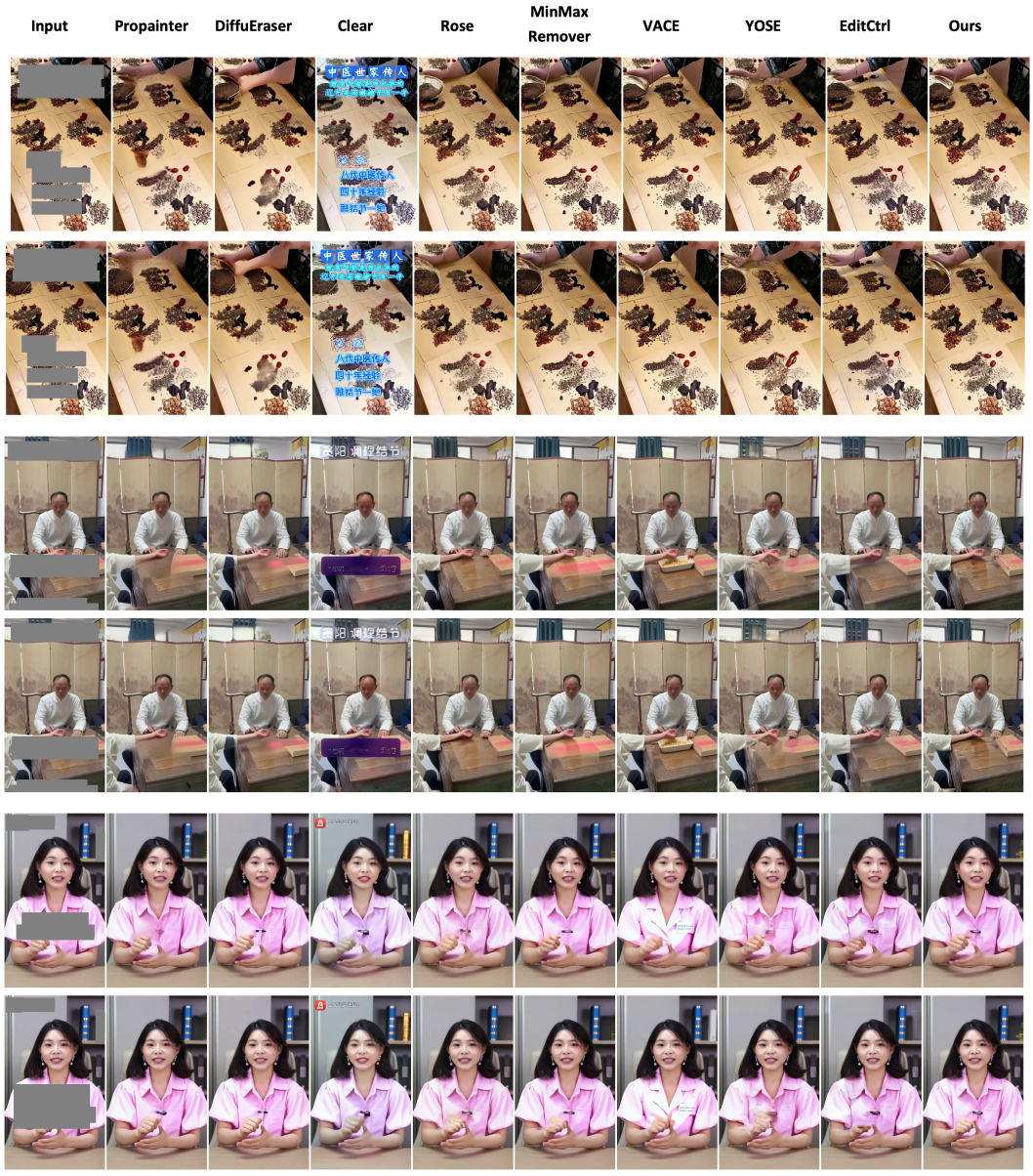


Figure 8: Additional text and logo removal comparisons (1/2): three cases with overlays on scene content. Two frames are shown per case. Gray regions in the input column indicate the removal areas; PACE is shown in the rightmost column.

Additional text and logo removal comparisons (continued).

Fig. 9 compares removal on largely uniform backgrounds. EditCtrl produces conspicuous text-like patterns or additional content in the removal regions. PACE produces cleaner black and blue backgrounds that blend with their surroundings. Taken together, the two groups show that PACE achieves better removal quality than ProPainter, DiffuEraser, and the lightweight methods, with results close to full-video methods in the examples shown.

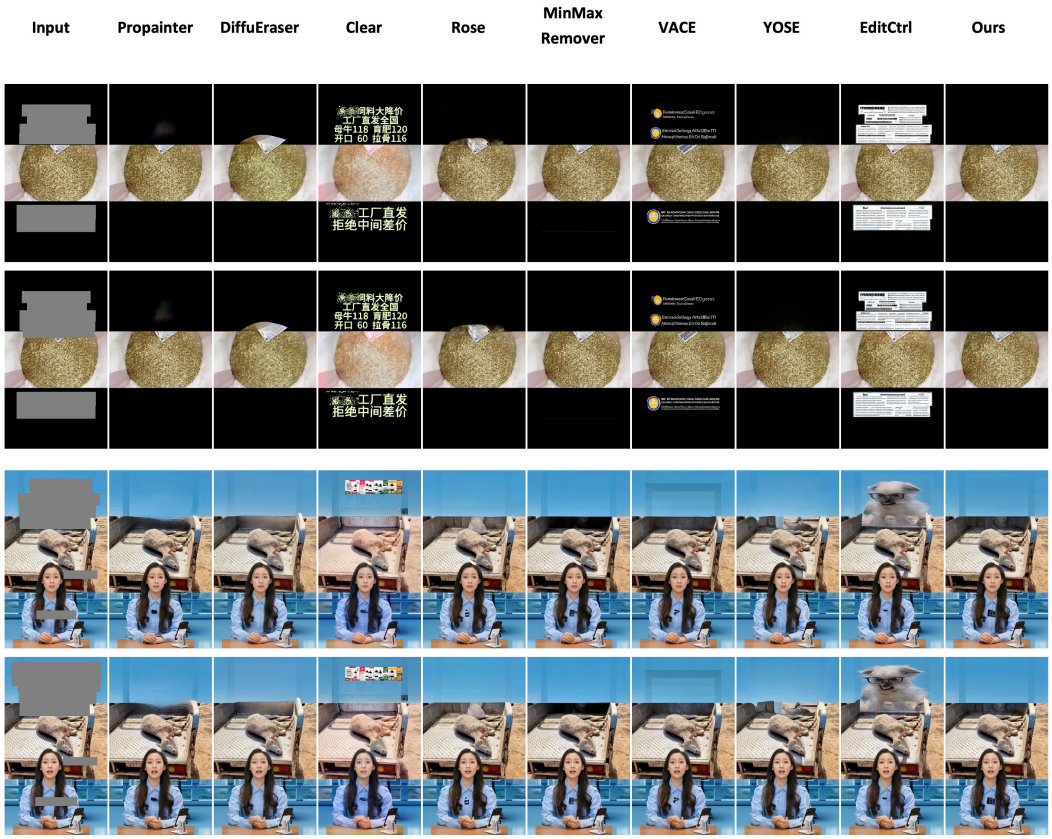


Figure 9: Additional text and logo removal comparisons (2/2): two cases with overlays on largely uniform backgrounds. Two frames are shown per case. Column order is identical to Fig. 8, with PACE in the rightmost column.