SEDiT: MASK-FREE VIDEO SUBTITLE ERASURE WITH A ONE-STEP DIFFUSION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing video subtitle removal methods inpaint masked frames and must first locate the subtitles. The accuracy of this segmentation limits the final quality. We present **SEDiT**, a one-stage video **S**ubtitle **E**rasure method built on a one-step **Di**ffusion **T**ransformer. SEDiT is mask-free *at inference*: it needs no mask, OCR, or segmentation at test time. During training, a coarse subtitle box is used only to weight the loss in subtitle regions.

We give a **conditional-mean characterization** of one-step inference: under the exact conditional velocity field, a single Euler step from pure noise returns the target's conditional mean. Its unconditional latent squared-error risk equals the expected trace of the target's conditional covariance. For any fixed learned field, the risk decomposes exactly into this irreducible uncertainty and an endpoint field-error term. We hypothesize that subtitle removal often has low conditional uncertainty, so this characterization is relevant to reconstruction rather than sampling from the conditional distribution. It does not guarantee perceptual quality. With chunk-wise streaming inference, SEDiT enables one-step erasure on native 1080p videos of minute-scale duration.

On our paired VSR-Bench-400 benchmark, SEDiT reaches 31.59 dB PSNR and 24.06 FVD. Its raw outputs have higher fidelity than those of the evaluated mask-based baselines, which receive ground-truth subtitle boxes rather than detected ones. SEDiT takes 4s per 65-frame 1080p clip—about $37\times$ faster than Minimax-Remover and $266\times$ faster than EffectErase—without step distillation. On 30 real videos of roughly two minutes, primarily from television dramas and short-form dramas, subtitles are completely removed in 25 videos by SEDiT and in 29 by SEDiT-Distill under our strict full-clip criterion. Supplementary videos provide further qualitative evidence on real videos.

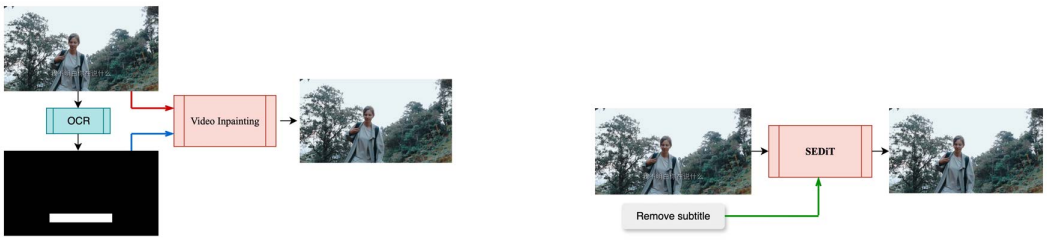
1 INTRODUCTION

Video diffusion models (Blattmann et al., 2023; Kong et al., 2024; HaCohen et al., 2024; Wan et al., 2025) have made major progress in video generation. Many recent models use the Diffusion Transformer (DiT) (Peebles & Xie, 2023) architecture to improve generation quality. These advances in video generation have also improved restoration quality in video inpainting, an important video editing task.

Subtitle removal is an important object-removal task. Generic object removal is usually interactive and applied to short, single-shot clips of about 10 seconds. Subtitle removal, in contrast, must process entire videos with multiple shots and no user interaction. Transparency, stylized fonts, and gradient effects often make subtitles hard to segment accurately. As illustrated in Figure 1a, most prior methods locate subtitles with bounding boxes from OCR. These boxes lose pixel-level information in the gaps between characters, which limits restoration quality. Fine-grained text masks require a separate text-matting model, making the system more complex.

To address these limitations, we propose a one-stage framework for video subtitle removal without OCR-based localization. As illustrated in Figure 1b, we remove the OCR module and use a predefined prompt as the editing instruction. Inspired by instruction-based image editing methods such as *FLUX.1 Kontext* (Labs et al., 2025) and *Qwen-Image* (Wu et al., 2025), we propose *SEDiT*, which

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.



(a) Two-stage subtitle erasure

(b) One-stage subtitle erasure

Figure 1: Previous mask-based subtitle removal frameworks compared with our mask-free, one-stage framework.

uses a fixed user prompt. This simple design suits video subtitle removal, which needs no user interaction and benefits from efficient batch processing. We use LTX-Video (HaCohen et al., 2024) as the backbone. Its VAE has a high spatio-temporal compression ratio, allowing us to process 1080p video directly.

Our main contributions are as follows.

- **Mask-free one-stage erasure.** SEDiT erases subtitles in a single stage, with no mask, OCR, or segmentation at test time. During training, a coarse subtitle box is used only to weight the loss in subtitle regions. Providing the reference video *in context* rather than by channel concatenation gives better results under the same training budget (Table 3). We also build a synthesis pipeline that renders subtitles with random fonts, layouts, and transitions to produce paired data at scale.
- **A conditional-mean characterization of one-step inference.** We show that one-step inference from pure noise under the exact field returns the conditional mean, with latent squared-error risk $\mathbb{E}_{\mathbf{z}_{\text{ref}}}[\operatorname{Tr}\operatorname{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{\text{ref}})]$. We also derive an exact orthogonal risk decomposition for any fixed learned field. This distinguishes Bayes-optimal latent reconstruction from sampling from the conditional distribution. It does not claim perceptual optimality or that one step is always sufficient.
- **Minute-scale, native-1080p deployment.** Chunk-wise streaming inference with single-frame overlap supports much longer videos. SEDiT processes a 65-frame 1080p clip in 4s ($37\times$ / $266\times$ faster than Minimax-Remover / EffectErase). On 30 real minute-scale videos it meets the strict full-clip removal criterion on 25 videos, and SEDiT-Distill on 29. Supplementary videos provide qualitative evidence on full sequences.
- **OOD mitigation using real failure cases.** As an optional training-time extension, we use Qwen3.8-27B to find potentially difficult cases in real user-generated videos where SEDiT may leave residual subtitles. A single-frame FLUX.2-klein-9B teacher provides additional distillation pairs, which are mixed with the original video data at a ratio of 6:4. SEDiT-Distill improves perceptual and distribution-level quality on the controlled benchmark while retaining the video-level training objective. The base SEDiT model and its main benchmark results are unchanged.

2 Related Work

Instruction-based image editing. InstructPix2Pix (Brooks et al., 2023) showed the potential of instruction-guided diffusion models for image editing. More recently, *FLUX.1 Kontext* (Labs et al., 2025) achieved strong results with a simple flow matching model. It concatenates context and instruction tokens *along the sequence dimension* and directly predicts velocity. *Qwen-Image* (Wu et al., 2025) follows a similar approach by concatenating the VAE-encoded input-image latent with the noisy image latent along the same dimension. Concatenating conditioning and noisy latents along the sequence dimension has become a common approach in image editing. This motivates our study of video inputs as conditions for video editing.

Diffusion-based video inpainting. Diffusion-based video inpainting is a form of masked video-to-video editing (MV2V) that requires a sequence of masks for the region of interest. Given a noisy

latent $\mathbf{z}_t$, a masked video latent $\mathbf{z}_m$, and the mask $\bar{\mathbf{m}}$, these methods complete the masked video with a conditional diffusion model $u_\theta(\mathbf{z}_t, \mathbf{z}_m, \bar{\mathbf{m}}, \text{prompt}, t)$. The three inputs are usually concatenated along the *channel* dimension. Both Minimax-Remover (Zi et al., 2025) and EraserDiT (Liu & Hui, 2025) use this approach. Minimax-Remover is built on Wan2.1-1.3B (Wan et al., 2025) and increases the number of backbone input channels from 16 to 48 to accommodate the extra inputs.

Video subtitle removal. Mask-free removal of burnt-in subtitles predates diffusion models. Blind video decaptioning methods (Kim et al., 2019; Chu et al., 2021) train an encoder-decoder to locate and inpaint text without a mask at test time. They use information from multiple frames to recover occluded content, but operate on short, low-resolution clips, and their model capacity limits background fidelity. More recently, CLEAR (He et al., 2026) revisits mask-free subtitle removal with an end-to-end model. Mask-free inference alone therefore does not distinguish SEDiT from prior work. Our contribution is to enable it on a modern video DiT at native 1080p resolution and minute scale with a single sampling step. We observe two practical limitations in CLEAR: a pronounced global color shift and failure to remove large-font subtitles or handle long sequences (Appendix E.2). In contrast, SEDiT keeps the pretrained backbone unchanged and adds only a LoRA adapter. This preserves the generative prior needed to reconstruct high-resolution backgrounds. SEDiT also inherits the base model's streaming behavior instead of being limited to a fixed clip length.

3 Methodology

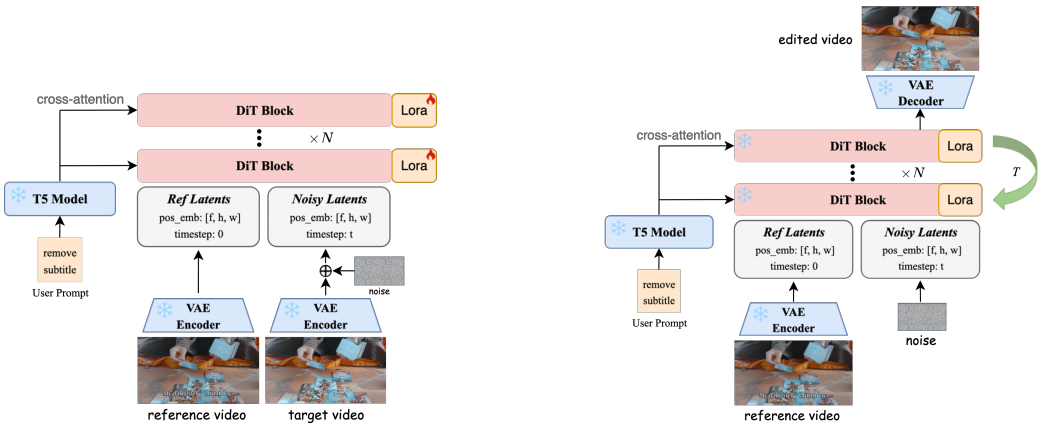
3.1 Preliminaries

Flow matching. Video generation models based on flow matching use a neural network to learn a time-dependent vector field. This field moves samples from a noise distribution toward the target video distribution. Given an input video $\mathbf{x} \in \mathbb{R}^{f\times h\times w\times c}$, a pretrained variational autoencoder (VAE) encoder $\mathcal{E}$ maps $\mathbf{x}$ to the latent representation $\mathbf{z_0} = \mathcal{E}(\mathbf{x})$. The rectified flow-matching loss is

$$\mathcal{L}_\theta = \mathbb{E}_{t\sim p(t),\mathbf{z}_0,\epsilon,c_{\text{text}}}\left[\left\|v_\theta\left(\mathbf{z}_t, t, \mathbf{z}_{\text{ref}}, c_{\text{text}}\right) - \mathbf{v}_t\right\|_2^2\right], \tag{1}$$

where $\mathbf{x}_0$ is the clean subtitle-free video, $\mathbf{x}_{\text{ref}}$ is the corresponding subtitle-rendered video, and $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ and $\mathbf{z}_{\text{ref}} = \mathcal{E}(\mathbf{x}_{\text{ref}})$ are their VAE latents. The latent $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\epsilon$ is linearly interpolated between $\mathbf{z}_0$ and noise $\epsilon \in \mathcal{N}(0, \mathbf{I})$, c_{text} is the text prompt, and the velocity is $\mathbf{v}_t = \frac{d\mathbf{z}_t}{dt} = \epsilon - \mathbf{z}_0$. We do not assume an additive decomposition in VAE latent space.

3.2 Model Architecture



(a) Training architecture.

(b) Inference architecture.

Figure 2: Overview of our mask-free video subtitle erasure framework. We add a conditioning video branch alongside the original video branch. The VAE encoder maps the conditioning video into tokens. These are concatenated with the video latent tokens and passed to the DiT.

Figure 2 shows the overall framework. We use LTX-Video-2B-0.9.6 (HaCohen et al., 2024), a Diffusion Transformer (DiT), as the base model and train it to generate videos conditioned on a text prompt and a reference video. The pretrained VAE encoder encodes the reference video with subtitles, and the resulting latent is divided into patches to form $\mathbf{z}_{\text{ref}} \in \mathbb{R}^{B\times(F\times H\times W)\times C}$, with the same shape as the noisy video tokens. The reference latents stay clean throughout denoising, so we set their denoising timestep to 0. We concatenate the reference and noisy video tokens along the sequence dimension to obtain $\mathbf{z}_{in} \in \mathbb{R}^{B\times 2(FHW)\times C}$. The DiT blocks process these tokens jointly. Here F, H, W denote the latent token-grid dimensions and C the latent channel dimension.

We encode positional information via 3D Rotary Positional Embedding (RoPE) (Su et al., 2024), aligning the embeddings of the reference $\mathbf{z}_{\text{ref}}$ with those of the noisy tokens $\mathbf{z}_{\text{noisy}}$: if a token position is the triplet $\mathbf{u} = (\mathrm{f}, \mathrm{h}, \mathrm{w})$, we set $\mathbf{u}_{\text{ref}} = \mathbf{u}_{\text{noisy}}$, which ensures frame-by-frame alignment between the reference and the target video.

Since subtitles occupy only a small portion of the video and the reference and target videos share identical positional encodings, directly applying Equation 1 can bias the model toward reconstructing the reference video. We therefore assign a higher loss weight to subtitle regions:

$$\mathcal{L}_{\text{weighted}} = \mathbb{E}\left[\|v_\theta - \mathbf{v}_t\|_2^2 \odot (\mathbf{I} + \alpha \mathbf{M}_{\text{subtitle}})\right], \tag{2}$$

where $\mathbf{M}_{\text{subtitle}}$ is the mask obtained by filling the subtitle bounding box, $\mathbf{I}$ is an all-ones tensor of the same shape, and α is a scalar. This is a purely spatial reweighting: it rescales the per-token regression error inside the subtitle box and leaves the objective otherwise unchanged.

3.3 Data Synthesis

For efficient training, we render subtitles that resemble those in real videos. Figure 4 (Appendix A) shows our data synthesis pipeline. It has four components: (1) collecting subtitle content by parsing real `*.srt` files in English, Chinese, Japanese, and Korean from films, anime, and television dramas; (2) setting font attributes by randomly choosing the typeface, size, bright or dark text tone, outline color and thickness, and shadow rendering; (3) choosing horizontal and vertical layouts with probabilities 75% and 25%; and (4) adding abrupt transitions and fade-in/fade-out effects, including time-synchronized color changes in MV-style lyrics. Appendix A lists the complete sampling ranges and probabilities for each component.

3.4 Real-failure-driven Off-policy Distillation

The rendered subtitle data used in the main training stage provides paired supervision at scale, but does not cover the full distribution of subtitles in real videos. To reduce this synthetic-to-real gap, we add an off-policy single-frame distillation branch based on real failure cases. We use the multimodal model Qwen3.8-27B (Qwen Team, 2026) to find candidate examples in user-generated videos where the base SEDiT output may contain residual subtitles. We sample frames at 1 fps and focus on subtitle styles and rendering conditions that are difficult for the base model. Qwen lets us find candidates at scale. Human confirmation is part of the separate evaluation protocol described below; we do not claim that all training examples were manually reviewed.

For each retained pair, we use the single-frame FLUX.2-klein-9B (Black Forest Labs, 2026) model as a teacher. We apply AdaIN-based color correction to reduce the teacher's global color shift. We then form a pseudo-target from the corrected teacher output inside the subtitle mask and the reference frame outside it. We retain pairs only when the shorter image side is at least 1024 pixels and the maximum mask ratio is at most 0.40. We cache each pair with its reference, pseudo-target, and difference image. To train the student, we mix the original video data with the single-frame distillation data at a ratio of $6{:}4$. This preserves the original pipeline's temporal supervision and video modeling while adding appearance statistics from real data and difficult subtitle cases. We call the resulting model SEDiT-Distill. This optional branch addresses subtitle distributions not covered by the procedural renderer; it does not replace the base SEDiT model or its inference architecture.

3.5 Long Video Inference

In practice, videos can be much longer than a single model window, especially in television dramas and short-form dramas. We first use the scene detection algorithm TransNetV2 (Soucek & Lokoc,

2024) to divide each video into shots, then process each shot separately. Short shots (≤ 5 seconds) are handled directly. Longer shots use the chunk-wise strategy in Figure 5, with chunk sizes set by resolution: 121 frames for 720p, 65 for 1080p, and 41 for 1440p. We use first-frame conditioning for long sequences (Figure 6). During training, we replace the first frame's noisy latent with its clean counterpart with probability 0.2 and set its diffusion timestep to zero. At inference, the first frame of each chunk after the first is initialized with the last frame of the previous chunk. One-frame overlap minimizes boundary warping error on the GT subset and gives near-best boundary-window flicker on real videos (Appendix B). To reduce memory use, we stream frame data to a TS (Transport Stream) video file after processing each shot.

To avoid writing complex prompts at inference, we use a predefined training prompt that does not specify the subtitle content: *"Please remove the subtitle text from the video while preserving the character appearance, background composition, and color style. Do not add any new elements."* The task is well defined and the prompt is fixed, so the prompt serves only as a required input to the base model and has little effect on subtitle erasure. By default, our method removes subtitles automatically without user interaction. If a user wants to limit the edit to a specific region, SEDiT can protect the remaining region in latent space, as described in Appendix G.9.

3.6 What Does One-Step Inference Compute?

Reconstructing a paired target and sampling from its full conditional distribution are different objectives. Omitting the fixed prompt from the notation, write $R = \mathbf{z}_{\text{ref}}$ and $v^*(\mathbf{z}_t, t \mid R) = \mathbb{E}[\epsilon - \mathbf{z}_0 \mid \mathbf{z}_t, R]$. The endpoint field characterizes reconstruction without assumptions on transport-path curvature.

Proposition 1 (Endpoint conditional mean)**.** *Assume $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is independent of $(\mathbf{z}_0, R)$ and $\mathbb{E}\|\mathbf{z}_0\|^2 < \infty$. Since $\mathbf{z}_1 = \epsilon$, the exact-field Euler step satisfies*

$$\begin{aligned} \hat{\mathbf{z}}_0^* &= \epsilon - v^*(\epsilon, 1 \mid R) = \mathbb{E}[\mathbf{z}_0 \mid R], \\ \mathbb{E}\|\hat{\mathbf{z}}_0^* - \mathbf{z}_0\|^2 &= \mathbb{E}_R[\operatorname{Tr}\operatorname{Cov}(\mathbf{z}_0 \mid R)]. \end{aligned} \tag{3}$$

Thus the exact step is Bayes-optimal for latent squared error, not a perceptual or pixel-space guarantee, and generally does not sample the conditional distribution. For a fixed trained network with $\mathbb{E}\|v_\theta(\epsilon, 1 \mid R) - v^*(\epsilon, 1 \mid R)\|^2 < \infty$, let $\hat{\mathbf{z}}_0^\theta = \epsilon - v_\theta(\epsilon, 1 \mid R)$. Orthogonality gives

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{z}}_0^\theta - \mathbf{z}_0\|^2 = \mathbb{E}_R[\operatorname{Tr}\operatorname{Cov}(\mathbf{z}_0 \mid R)] \\ + \mathbb{E}\|v_\theta(\epsilon, 1 \mid R) - v^*(\epsilon, 1 \mid R)\|^2. \end{aligned} \tag{4}$$

Appendix F proves both identities, including the general starting-time case. Spatially weighted training does not invalidate this decomposition for a fixed field, but does not guarantee a small endpoint error; a time-averaged training loss alone does not control $t = 1$.

For subtitle erasure, visible context and adjacent frames motivate the *hypothesis* of low conditional uncertainty (Appendix G). If both terms in Equation 4 are small, one-step latent reconstruction is accurate. The step-count ablations are consistent with this hypothesis, but do not prove it causally (Appendix F.5); neither multimodality nor edit extent alone determines the required number of steps.

4 Experiments

4.1 Datasets

We collect 400K high-definition, subtitle-free videos from Pexels[1] as ground truth (GT). We add subtitles on the fly using the synthesis pipeline in Section 3.3. Because no public benchmark exists for video subtitle removal, we build the Video Subtitle Removal Benchmark (VSR-Bench-400). It contains 400 samples, each with a clean video, a subtitle mask, and a version with rendered subtitles. To avoid data leakage, its source videos do not overlap with the training videos. The 720p clips contain 121 frames and fit the 720p chunk size in one forward pass. The 1080p clips contain 81 frames, exceeding the 65-frame 1080p chunk size (Section 3.5), so we process them as two chunks with single-frame overlap. All fidelity metrics are computed over the full clip, so the 1080p results already include one chunk transition.

[1] https://www.pexels.com/

Table 1: Quantitative results on VSR-Bench-400, with full-frame and region-resolved metrics for the same raw model outputs. "Time" is the average processing time for one 1920×1080, 65-frame chunk on an A800 without acceleration. Full-clip fidelity uses the 81-frame 1080p and 121-frame 720p protocols. Mask-based methods receive **ground-truth box masks**, formed by filling the subtitle bounding box. Their localization is thus exact at the box granularity that an OCR pipeline would provide. The band is split into *stroke* pixels, where the subtitled input differs from the clean target, and the remaining box pixels (*band*$\backslash$*str*). Stroke pixels make up 30.9% of the box and 2.4% of the frame; Appendix G.6 gives the exact criterion. Here "out" denotes everything outside the box and "bd" the boundary ring. LPIPS is reported on the box crop, where it is well defined, rather than on scattered stroke pixels. Region metrics are averaged over the 400 clips, except the stroke column, which uses the 399 clips containing stroke pixels. SEDiT-Distill differs from SEDiT only in training, so its inference cost is identical. A dash means that a region-resolved result is not reported for that configuration. *VAE round-trip* is a reconstruction reference, not a competing method.

	Full-frame					Region-resolved				
Method	**PSNR** ↑	**SSIM** ↑	**LPIPS** ↓	**FVD** ↓	**Time** ↓	**PSNR**$_{\text{stroke}}$ ↑	**PSNR**$_{\text{band}\backslash\text{str}}$ ↑	**PSNR**$_{\text{out}}$ ↑	**LPIPS**$_{\text{in}}$ ↓	**LPIPS**$_{\text{bd}}$ ↓
EffectErase (50-step)	25.38	0.8160	0.1320	66.02	1064s	20.52	21.73	26.49	0.2341	0.2283
Propainter	26.82	0.8231	0.1778	65.00	38s	20.91	22.06	28.47	0.3596	0.3403
DiffuEraser	27.51	0.8146	0.1223	56.77	166s	21.20	22.26	29.35	0.2807	0.2682
Minimax-Remover (6-step)	28.31	0.8785	0.1011	39.35	150s	21.00	21.79	30.91	0.3172	0.3033
SEDiT (4-step)	31.30	0.8783	0.1001	26.76	8s	–	–	–	–	–
SEDiT (2-step)	31.60	0.8805	0.0982	24.91	6s	–	–	–	–	–
SEDiT (1-step)	31.59	0.8805	0.0981	24.06	4s	28.16	31.01	31.89	0.1195	0.1180
SEDiT-Distill (1-step)	31.52	0.8805	0.0972	21.52	4s	28.03	30.86	31.83	0.1131	0.1122
VAE round-trip (reference)	32.94	0.9002	0.0780	13.31	–	32.96	33.51	32.67	0.0679	0.0692

4.2 Implementation Details

We fine-tune LTX-Video-0.9.6 (2B) with Low-Rank Adaptation (LoRA) of rank 256, adding 381M trainable parameters (19% of the base model). Training runs for 15K iterations on 8 NVIDIA A800 (80GB) GPUs using AdamW, with batch size 32 and initial learning rate 2×10^{-4}. Inputs contain 65 frames at resolutions near 1080×1920 and 121 frames near 1280×720. We set α to 5 in Equation 2. At inference, the base SEDiT model uses one sampling step. SEDiT-Distill also uses the training-time branch in Section 3.4; neither model uses inference-time model distillation. Table 1 provides empirical support for the one-step setting. The strong video conditioning and observed *copy-preservation behavior* motivate, but do not establish, the low-uncertainty hypothesis in Section 3.6.

4.3 Quantitative Experiments

We compare our method with Minimax-Remover (Zi et al., 2025), a leading open-source DiT-based method for video object removal; the GAN-based Propainter (Zhou et al., 2023); the diffusion-UNet-based DiffuEraser (Li et al., 2025); and EffectErase (Fu et al., 2026), a recent multi-step model for video effect removal evaluated with its default 50 sampling steps. For mask-based methods, we fill the subtitle boundaries to obtain rectangular masks (ground-truth box masks). The masked input ($V_{\text{masked}} = V \cdot (\mathbf{1} - \mathbf{M}) + \mathbf{0} \cdot \mathbf{M}$) contains no visible subtitles and is independent of the subtitle rendering process. We measure frame- and clip-level fidelity with PSNR, SSIM, LPIPS, and FVD. As shown in Table 1, SEDiT with 1 and 2 sampling steps gives the strongest reported raw-output results on the main reconstruction metrics among the evaluated methods, while being an order of magnitude faster. It completes a native 1080p clip in 4s, roughly $266\times$ faster than EffectErase (1064s) and $37\times$ faster than Minimax-Remover (150s). Among our configurations, $step = 2$ gives the highest PSNR (31.60), while $step = 1$ gives the best LPIPS and FVD at identical SSIM. We choose $step = 1$ for its efficiency.

VAE reconstruction reference. In the last row of Table 1, we pass the ground truth through the VAE using the same chunking as at inference, but without the DiT. This gives an empirical reconstruction reference for the fully re-encoded pipeline. SEDiT (1-step) is within 1.35 dB PSNR and 10.75 FVD of this reference. We use PSNR and FVD as the main full-frame indicators because the subtitle band covers only a small fraction of each frame. Full-frame SSIM and LPIPS therefore mainly reflect preservation of the unedited region rather than erasure quality, as quantified in Appendix G.5.

Region-resolved fidelity. Table 1 splits each frame into subtitle stroke pixels, the remaining pixels in the subtitle band, and everything outside the band. On stroke pixels, which every method must

synthesize, SEDiT reaches 28.16 dB versus 21.20 dB for the best baseline, a margin of 6.96 dB. On the rest of the band, which the baselines must regenerate only because their mask is a box, the margin is 8.74 dB (31.01 versus 22.26). The full-frame numbers therefore *understate* the gap by more than half (+3.28 dB). This split separates the two pixel classes, but not the input conditions: a finer mask would also give the baselines in-band context that was unavailable when producing the outputs scored here (Appendix G.6). Outside the band, SEDiT has the best raw-output score at 31.89 dB, 0.98 dB above Minimax-Remover and within 0.77 dB of the reconstruction reference; every method is scored on its raw full-frame output, so a mask-based pipeline could trivially score higher in this column by compositing its result onto the unedited input. In-band LPIPS is 0.1195 versus 0.2341 for the best baseline, a factor of 2.0. On the boundary ring, SEDiT's score of 0.1180 is indistinguishable from its in-band value, indicating no halo specific to the mask edge, while every baseline is 1.9–2.9× worse. The gaps to the reconstruction reference are 0.77 dB outside the band, 2.51 dB on non-stroke band pixels, and 4.80 dB on stroke pixels, so the remaining reconstruction gap is concentrated where the glyphs were.

Results of off-policy distillation. Table 1 also reports the distilled model. Compared with SEDiT, SEDiT-Distill is slightly lower on every PSNR column (for example 28.03 vs. 28.16 dB on stroke pixels and 31.52 vs. 31.59 dB full-frame) at unchanged SSIM, and better on the perceptual measures: in-band LPIPS improves from 0.1195 to 0.1131, boundary LPIPS from 0.1180 to 0.1122, full-frame LPIPS from 0.0981 to 0.0972, and FVD from 24.06 to 21.52. Single-frame distillation from real failure cases therefore improves perceptual quality at boundaries and video-level distributional fidelity without materially changing reconstruction fidelity. The branch is used only during training, so it adds no inference-time cost.

Real minute-scale videos. Beyond the rendered benchmark, we evaluate on 30 real videos of roughly two minutes each, from short-form dramas, independent creators, and television dramas on the Internet, in both horizontal and vertical formats. They do not overlap with the training data of SEDiT or SEDiT-Distill, including the real videos used for off-policy distillation. We screen outputs for residual subtitles with Qwen3.8-27B under a fixed prompt and seed $= 42$, and verify every count by frame-by-frame human inspection of the full clip (Appendix G.1). Table 2 reports the result: neither model leaves a residual on any VSR-Bench-400 clip ($0/400$), while on the real videos $5/30$ still show residuals for SEDiT and $1/30$ for SEDiT-Distill. The same videos give the usability rate of Section 4.5. They were selected to include difficult subtitle styles, so these counts describe a failure-enriched set rather than the overall real-world failure rate.

Table 2: Results on the controlled benchmark and on the same set of 30 real minute-scale videos. Qwen3.8-27B screens outputs with a fixed prompt and seed $= 42$, and all counts were verified by frame-by-frame human inspection of the full clips. A video is usable only if no residual remains anywhere in it, so the usability rate is the complement of the residual count.

Data	Measure	SEDiT	SEDiT-Distill
VSR-Bench-400 (400 clips)	clips with residual	0/400 (0.00%)	0/400 (0.00%)
Real minute-scale (30 videos)	videos with residual	5/30 (16.7%)	1/30 (3.3%)
Real minute-scale (30 videos)	usable videos	25/30 (83.3%)	29/30 (96.7%)

Human side-by-side comparison (GSB). We run a blind side-by-side study on the 400 VSR-Bench-400 clips with 20 human participants. The outputs of SEDiT (1-step) and Minimax-Remover (6-step) are shown together with random left/right placement, and raters are not told which method produced either output. Each participant rates 20 clips drawn at random without replacement, so the 20 participants together cover every clip exactly once. The criterion is simple: whether subtitles are cleanly removed, rather than whether the output meets a broad aesthetic standard. Each rater scores both outputs on a 5-point scale focused on visible residuals and restoration damage, and we *derive* a Good/Same/Bad (GSB) judgment per clip from the two scores of that same rater (Appendix G.1). The result is $G{:}S{:}B = 336{:}64{:}0$: SEDiT scores higher on 336 of the 400 clips, ties on 64, and never scores below the baseline, giving a GSB ratio of $(G - B)/(G + S + B) = 84.0\%$ with an uncorrected two-sided sign test at $p < 10^{-6}$. Since each clip is rated once, this measures relative erasure quality per clip rather than inter-rater reliability or absolute quality; absolute quality is reported as the usability rate in Section 4.5. Figure 3 shows the two failure modes behind this margin: residual glyphs where the mask is incomplete and blur around the mask.

4.4 Qualitative Experiments on Real Videos

In addition to the paired quantitative evaluation on VSR-Bench-400, we provide qualitative examples from real film and television videos, mainly television dramas and short-form dramas. These videos are not part of VSR-Bench-400. They contain subtitle rendering, scene motion, shot transitions, and video conditions that our synthetic pipeline does not fully represent. Our method removes subtitles across languages and styles, including stylized and blurred subtitles that challenge conventional OCR, Russian subtitles absent from the training data, and large bordered subtitles that cover parts of faces. It also leaves subtitle-free frames unchanged. Appendix C discusses each multilingual example (Figure 7).



(a) Focus on the yellow boxes.



(b) Focus on the white boxes.

Figure 3: Visual comparison with state-of-the-art (SOTA) methods. The green region marks the subtitle mask obtained using SAM2. *Best viewed when zoomed in.*

To compare our method with state-of-the-art mask-based video inpainting under real-world conditions, we locate subtitles through an interactive interface, track them with SAM2 (Ravi et al., 2024), and pass the masks to Minimax-Remover (Zi et al., 2025). This segmentation-based strategy gives more precise subtitle regions than OCR-based boxes and therefore favors the mask-guided baseline. In Figure 3a, our mask-free method reconstructs occluded body parts of distant characters, whereas the baseline produces noticeable blur. The final column shows that the baseline can still give reasonable reconstructions under mild occlusion. Figure 3b shows error propagation from inaccurate masks: an incomplete mask leaves subtitle residuals, and the error persists even when a later input frame has no subtitles. **Appendix D provides additional high-resolution results on VSR-Bench-400 (Figures 8–10). Supplementary videos show real-video examples outside this benchmark.**

4.5 Long-video evaluation

To our knowledge, no existing subtitle-removal method handles minute-scale sequences directly, which makes a frame-level comparison on long videos hard to interpret: the visual quality of CLEAR (He et al., 2026) drops as the sequence grows, so per-frame fidelity scores no longer tell us whether the result is usable. We therefore report a *usability rate* on the 30 real minute-scale videos of Section 4.3, using a strict binary criterion: a video is usable only if all subtitles are removed throughout, and any residual at any time makes it unusable. Three annotators independently inspect each full video. SEDiT is usable on 25 of 30 videos (83.3%) and SEDiT-Distill on 29 of 30 (96.7%), matching the residual counts in Table 2; the one remaining failure of SEDiT-Distill is an occasional residual on a small caption. This is initial evidence that chunk-wise streaming inference (Section 3.5) is usable at minute scale for television dramas and short-form dramas, and that the distillation branch closes most of the gap on this set, rather than exhaustive evidence across all real-world video categories.

End-to-end cost. The "Time" column of Table 1 measures only the model. Running the complete pipeline on these videos at 1080p and 25 fps – shot detection, decoding, inference over all chunks, and streaming output to TS – gives a real-time factor (RTF) of 5.5 on a single A800 in bf16, where RTF is wall-clock time divided by video duration, so a 40-minute episode takes roughly 3.7 GPU-hours. Even the full pipeline is an order of magnitude cheaper than running the models alone for the strongest baselines, whose algorithm-only timings correspond to an RTF of 57.7 for Minimax-Remover and 409 for EffectErase. Appendix G.4 gives the cost breakdown and an optional variant that processes only the lower half of each frame.

4.6 Ablation study

Table 3: Quantitative comparison of conditioning methods on VSR-Bench-400 under an equal training budget, both with 1-step inference.

Conditioning	Training steps	LoRA rank	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓
In-context	3,000	256	31.30	0.8784	0.0998	26.43
Channel-concatenation	3,000	256	27.82	0.8392	0.1247	51.89

Effectiveness of subtitle-region weighting. Setting $\alpha = 0$ in Equation 2 gives the unweighted conditional rectified flow-matching loss. As shown in Figure 12 (Appendix G.3), this does not completely remove subtitles with complex styles such as gradient-colored text, whereas subtitle-region weighting focuses capacity on subtitle regions and improves removal.

Comparison of conditioning methods. Table 3 shows that in-context conditioning outperforms channel concatenation on all four metrics. Both use the same 1-step sampling and training budget, so the gap is due to the conditioning scheme rather than the sampling procedure. This supports the in-context design choice. It is not evidence for the one-step analysis in Section 3.6, which concerns a task property and does not depend on how the condition is provided.

5 Limitations

SEDiT-Distill is an optional training-time extension that addresses failures of the base model. Mining real failure cases targets difficult subtitle styles and rendering conditions that leave residual glyphs. Mixing the distilled data with the original video data at a $6{:}4$ ratio retains video-level temporal supervision. On the 30 manually verified real minute-scale videos, the residual count falls from $5/30$ for SEDiT to $1/30$ for SEDiT-Distill. This result is encouraging, but the set was selected to include difficult subtitle styles, so it does not establish broad OOD generalization.

The remaining empirical limitation is the lack of a broad, carefully annotated real-video benchmark. Our quantitative benchmark uses procedurally rendered subtitles over clean videos, the real-video evaluation covers 30 minute-scale videos, and these videos have no clean frame-level ground truth. Video-level detections, the usability rate, and visual examples therefore complement, but do not replace, controlled paired evaluation. Since the usability criterion counts any residual as a failure, it also says nothing about background damage or temporal artifacts. The planned 50-video evaluation in Appendix G.2 would provide stronger evidence for OOD mitigation, but being built from failure cases it would not estimate the population-wide residual rate. Finally, Section 3.6 characterizes conditional-mean reconstruction under the exact field and the risk of a fixed learned field. It guarantees neither a small endpoint fitting error nor perceptual quality. Low conditional uncertainty for subtitle erasure remains a hypothesis, not a directly measured property, although we measure the one-step estimator's seed sensitivity (Appendix F.5).

6 Conclusion

We propose **SEDiT**, a one-stage, mask-free framework for high-fidelity video subtitle erasure. It requires no mask, OCR, or segmentation at test time. During training, a coarse subtitle box is used only to weight the loss in subtitle regions. Keeping the base architecture unchanged enables efficient LoRA fine-tuning. One-step inference with chunk-wise streaming removes subtitles from minute-scale, native-1080p videos at 4s per 65-frame clip. Our conditional-mean characterization shows what one-step inference computes under the exact field. The orthogonal risk decomposition separates irreducible conditional uncertainty from learned endpoint error. This reconstruction-based account is consistent with the step-count results, but does not establish low conditional uncertainty or guarantee perceptual quality.

AI USE STATEMENT

We used a large language model to retrieve and find information, such as related work, and to polish text written by the authors. The authors checked all retrieved references and all polished passages and take full responsibility for the paper's content.

REFERENCES

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025.

Black Forest Labs. Flux.2-klein-9b. Hugging Face model card, 2026. URL `https://huggingface.co/black-forest-labs/FLUX.2-klein-9B`. Accessed 2026-09-17.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Pengpeng Chu, Weize Quan, Tong Wang, Pan Wang, Peiran Ren, and Dong-Ming Yan. Deep video decaptioning. In *BMVC*, pp. 273, 2021.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, et al. Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*, 2025.

Yang Fu, Yike Zheng, Ziyun Dai, and Henghui Ding. Effecterase: Joint video object removal and insertion for high-quality effect erasing. *arXiv preprint arXiv:2603.19224*, 2026.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

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. *arXiv preprint arXiv:2603.21901*, 2026.

Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep blind video decaptioning by temporal aggregation and recurrence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4263–4272, 2019.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

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

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-edit: Versatile video editing via instruction and reference guidance. *arXiv preprint arXiv:2603.02175*, 2026.

Jie Liu and Zheng Hui. Eraserdit: Fast video inpainting with diffusion transformer model. *arXiv preprint arXiv:2506.12853*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Qwen Team. Qwen3.8-27b. Hugging Face model card, 2026. URL `https://huggingface.co/Qwen/Qwen3.8-27B`. Accessed 2026-09-17.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11218–11221, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pp. 402–419, 2020.

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.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10477–10486, 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. *arXiv preprint arXiv:2505.24873*, 2025.

A Data Synthesis Details

This section gives the full sampling configuration for the four components of the data synthesis pipeline summarized in Section 3.3 and illustrated in Figure 4.

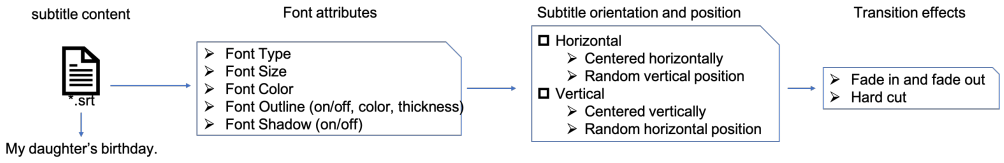


Figure 4: Data synthesis pipeline.

Subtitle content acquisition. We collected subtitle files (*.srt) from the Internet across several genres, including films, anime, and television dramas. The dataset covers English, Chinese, Japanese, and Korean. We use the Python library *pysrt* to extract and parse the subtitle content.

Font attribute configuration. We collected a set of widely used Chinese and English fonts, including *Source Han Sans*, *Source Han Serif*, *Alibaba PuHuiTi*, *Arial*, and *Helvetica*. The font size is sampled uniformly from $[40pt, 150pt]$. Text colors fall into two groups: bright and dark. For bright text, each of the red, green, and blue channels is sampled independently from $[231, 255]$, which gives light colors; for dark text, all three channels are sampled from $[0, 230]$, which gives darker ones. Bright text is given a dark outline with RGB values from $[0, 15]$, and dark text a light outline with RGB values from $[231, 255]$. With probability 50% the outline width is zero, so no outline is

visible; otherwise it is sampled uniformly from $[2pt, 5pt]$. Some subtitles are drawn with a shadow. To copy this effect, we shift the text slightly and draw it in a light black color behind the main text.

Subtitle layout specification. Subtitles are typically arranged horizontally near the bottom of the video frame, in both landscape and portrait videos. However, vertical subtitles also occur. We therefore assign probabilities of 75% and 25% to horizontal and vertical layouts, respectively. Horizontal subtitles are centered along the horizontal axis with a random vertical position. Vertical subtitles are centered along the vertical axis with a random horizontal position.

Transition effects for subtitle switching. Subtitle transitions in videos usually occur in two ways: (1) abrupt switching and (2) fade-in/fade-out effects. To simulate the latter, we change text opacity over time to create a smooth transition. In music videos (MVs), lyric text can also change color over time. We reproduce this behavior to cover lyric erasure in MV-style videos.

B Chunk-wise Long-video Inference

Figure 5 illustrates the chunk-wise inference strategy of Section 3.5, and Figure 6 the first-frame conditioning used to carry context across chunks.

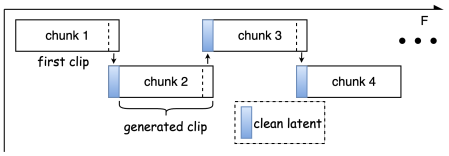


Figure 5: Long video inference strategy.

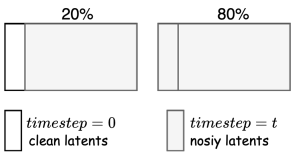


Figure 6: First-frame conditioning.

How much should chunks overlap? Because the video VAE compresses time by a factor of 8, the overlap is specified in *latent* frames: an overlap of k latent frames conditions the next chunk, at latent index 0 with strength 1, on the last $(k-1)\cdot 8+1$ output pixel frames of the previous chunk, which the VAE encodes to exactly k latent frames. Thus $k \in \{0,1,2,4\}$ corresponds to $\{0,1,9,25\}$ conditioning pixel frames, with $k=0$ meaning independent chunks. The pinned frames are dropped from the new chunk and replaced by the previous chunk's output, so every frame appears exactly once and a single seam sits at a fixed output index S.

We measure this on 150 clips with clean ground truth, each forced into exactly two chunks with the seam at $S=41$ so that one transition is isolated (1056×1920, 25 fps, 65 or 121 frames). Optical flow is estimated with RAFT-large (Teed & Deng, 2020) *on the clean ground truth only*, in both directions, never on the model outputs, which may themselves contain artifacts. A non-occlusion mask follows a forward–backward consistency check,

$$V_t = \Big[\, \|f_{t\to t-1} + \tilde{f}_{t-1\to t}\|^2 \le 0.01\big(\|f_{t\to t-1}\|^2 + \|\tilde{f}_{t-1\to t}\|^2\big) + 0.5 \,\Big], \tag{5}$$

where $\tilde{f}_{t-1\to t}$ is the backward flow warped into frame t, and the warping error of an output O is

$$E_{\text{warp}} = \frac{1}{T-1}\sum_{t=2}^{T} \frac{\big\| O_t - \mathcal{W}_{f_{t\to t-1}}(O_{t-1}) \big\|_1 \odot V_t}{\|V_t\|_1}, \tag{6}$$

with $\mathcal{W}$ a bilinear backward warp. We report E_{warp} separately on the boundary pair (O_{S-1}, O_S) – the exact seam pair, not a window around it – and on the interior pairs, together with the raw flicker $\frac{1}{HWC}\|O_t - O_{t-1}\|_1$.

Table 4 reports the result. The interior warping error stays close to 26×10^{-3} across all four settings. This acts as a control: the overlap affects only the seam, so the comparison is not driven by differences in overall video difficulty. Without overlap, the boundary error is $2.28\times$ the interior level; a single latent frame cuts that ratio to 1.50 and brings boundary flicker to within 4% of the interior value. Overlaps of 2 and 4 latent frames give no further gain and are in fact slightly worse, for two reasons that point the same way: training pins exactly one first-frame latent (Section 3.5),

so $k = 1$ matches the trained conditioning pattern while larger k does not, and in this fixed-seam design a wider overlap makes the second chunk longer, which puts large k at a disadvantage. We therefore use a single latent frame, the best of the settings we tested, while noting that boundary warping error is still $1.50\times$ the interior level, so the transition is reduced rather than removed.

Table 4: Boundary versus interior temporal consistency as a function of the overlap k, in latent frames, on the 150-clip ground-truth subset. Values are $\times 10^{-3}$ and lower is better; "ratio" is boundary over interior warping error. The interior column barely changes with the overlap and serves as a control.

		Warping error $E_{\text{warp}}\downarrow$			**Flicker $\downarrow$**	
Overlap k	**pinned frames**	boundary	interior	ratio	boundary	interior
0	0	59.57	26.18	2.28	29.70	22.18
1 (ours)	1	39.79	26.50	1.50	23.14	22.24
2	9	50.12	26.37	1.90	26.18	22.23
4	25	52.00	26.42	1.97	27.12	22.31

Overlap ablation on real videos. We additionally evaluate 30 real videos per overlap setting without clean ground truth (Table 5). Flicker is the mean absolute difference between adjacent output frames in normalized RGB, $\frac{1}{HWC}\|O_t - O_{t-1}\|_1$, with RGB values in $[0, 1]$. We compute boundary-window and non-boundary flicker for each video and then average across videos with equal weight, rather than pooling all frame pairs across videos. Unlike the exact seam-pair evaluation above, this evaluation uses a window around each boundary. Since these videos lack clean ground truth, we do not compute the GT-flow-based warping error defined above. Raw flicker includes changes due to scene motion and appearance as well as potential temporal artifacts; it is not a motion-compensated error.

Table 5: Overlap ablation on 30 real videos, with equal weight per video. Overlap denotes the actual number of pixel frames, not the script setting: `ov2` and `ov4` correspond to 9 and 25 frames, respectively. Flicker and its difference are multiplied by 1000; the ratio is dimensionless. Δ is boundary-window minus non-boundary flicker. No warping error is reported because clean GT is unavailable.

Overlap (frames)	**Videos**	**Boundary-window flicker $\downarrow$**	**Non-boundary flicker**	**$\Delta\downarrow$**	**Ratio**
0	30	23.384	17.856	5.528	1.310
1 (ours)	30	20.419	17.929	2.490	1.139
9 (`ov2`)	30	20.292	17.989	2.304	1.128
25 (`ov4`)	30	24.470	17.975	6.495	1.361

On these real videos, one-frame overlap reduces boundary-window flicker from 23.384 to 20.419, while nine-frame overlap attains the lowest value, 20.292, only 0.127 lower on the displayed scale. Non-boundary flicker remains similar across settings (17.856–17.989), whereas 25-frame overlap increases boundary-window flicker to 24.470. Thus the real-video results support one-frame overlap as a small-overlap choice with boundary-window flicker close to the best observed value, not as a universally optimal setting. The ranking differs from the GT-based seam-pair experiment above; the two evaluations use different videos and boundary measurements, and no statistical significance is claimed for the small difference between one- and nine-frame overlap.

C Multilingual Qualitative Results

As shown in Figure 7, the proposed method removes subtitles cleanly. In the third column of the top-left example, it removes text with strong subtitle-specific visual effects without visible traces. Such stylized and blurred subtitles are often difficult for conventional OCR systems, which may have low recognition accuracy or miss detections.

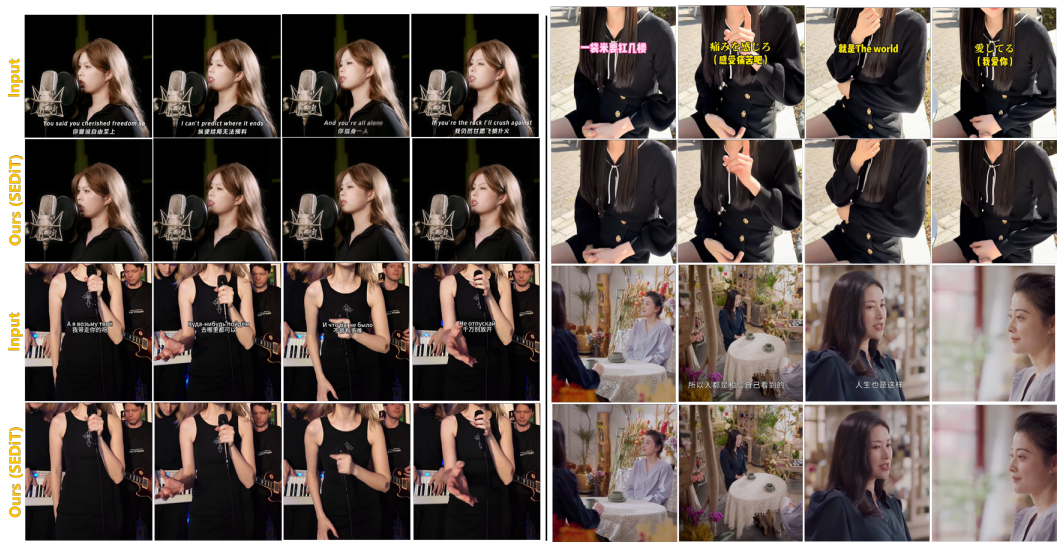


Figure 7: The evaluation results on multilingual subtitles include cases with mixed Chinese-English subtitles, mixed Chinese-Russian subtitles, Japanese subtitles, and Chinese-only subtitles. *Best viewed when zoomed in.*

The bottom-left example shows the robustness of our model: although Russian subtitles were not included in the training data, the method still removes them well. The top-right case shows that our method also handles colorful, bordered subtitles that cover a large part of the frame and obscure faces.

In the bottom-right example, the model restores complex textures such as patterned tablecloths. For frames without subtitles, our method detects the absence of text and avoids unnecessary changes. Overall, the results are visually coherent and contain few artifacts.

D Additional Visual Comparisons on VSR-Bench-400

This section provides the high-resolution visual comparisons referenced in Section 4. Figures 8–10 cover three groups of VSR-Bench-400 samples with different subtitle styles, fonts, and background complexity. In all cases, the green region marks the ground-truth (GT) subtitle mask obtained by filling the subtitle boundaries. This mask is supplied only to the mask-based baselines and is never used by SEDiT at inference.

E One-step Sampling Analysis

E.1 Comparison of Different Inference Steps

Video subtitle removal is a strongly constrained video editing task in which the input and target videos differ only slightly. Our experiments show that satisfactory results can be achieved with few sampling steps at inference. We therefore study how the number of sampling steps affects subtitle removal. As illustrated in Figure 11, using 1, 2, or 4 steps has little effect on the visual results. The $step = 1$ setting restores complex clothing textures well.

Table 6 provides additional comparisons, including 30-step SEDiT, Kiwi-Edit (Lin et al., 2026) (a general video editing model), and CLEAR (He et al., 2026) (a video subtitle removal model).

We observe that:

- 30-step sampling is worse than 1-step on every reported metric.
- In particular, FVD increases from 24.06 at 1 step to 26.76 at 4 steps and 31.76 at 30 steps, while PSNR decreases from 31.59 to 30.30.



Figure 8: Visual comparison on the VSR-Bench-400 dataset (group 1 of 3). The green highlighted region represents the subtitle mask, which is the ground-truth (GT) mask generated by filling the subtitle boundaries. *Best viewed when zoomed in.*

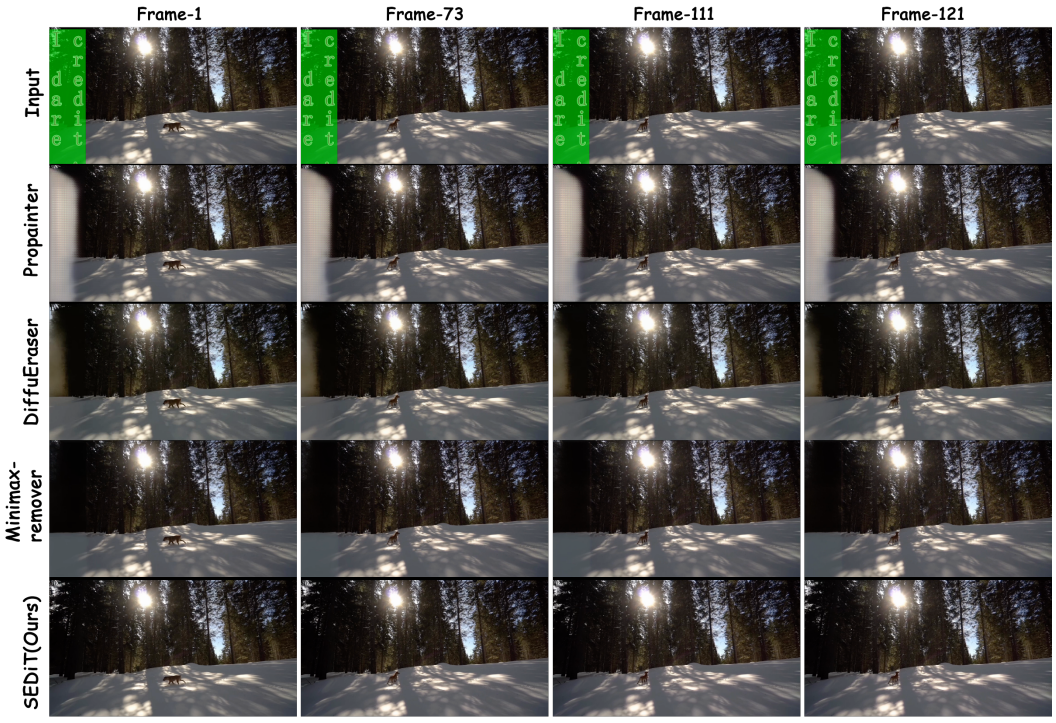


Figure 9: Visual comparison on the VSR-Bench-400 dataset (group 2 of 3). The green highlighted region represents the subtitle mask, which is the ground-truth (GT) mask generated by filling the subtitle boundaries. *Best viewed when zoomed in.*



Figure 10: Visual comparison on the VSR-Bench-400 dataset (group 3 of 3). The green highlighted region represents the subtitle mask, which is the ground-truth (GT) mask generated by filling the subtitle boundaries. *Best viewed when zoomed in.*



Figure 11: Comparison of visual effects under different sampling steps.

Table 6: Quantitative comparison of video editing methods.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓
SEDiT (1-step)	31.59	0.8805	0.0981	24.06
SEDiT (4-step)	31.30	0.8783	0.1001	26.76
SEDiT (30-step)	30.30	0.8626	0.1072	31.76
Kiwi-Edit (1-step)	29.02	0.8739	0.1038	34.33
Kiwi-Edit (30-step)	28.66	0.8725	0.1045	31.47
CLEAR (5-step)	19.39	0.7182	0.2379	167.66

- Kiwi-Edit shows a related step-count sensitivity, although its FVD does not follow the same monotonic trend.

This ordering is consistent with the reconstruction-based account of Section 3.6. Under the exact conditional field, the one-step estimator is the conditional mean $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{\text{ref}}]$ (Proposition 2), which minimizes latent squared error, not FVD or decoded-image error. The ablations do not isolate conditional uncertainty, endpoint fitting error, or errors along a multi-step trajectory. In particular, the FVD trend is an empirical observation, not a consequence of the risk identity or a causal proof that more steps accumulate error. These results are consistent evidence for the low-uncertainty hypothesis, not a direct measurement of the true conditional variance.

E.2 Evaluation configuration for CLEAR

CLEAR is evaluated with its official open-source implementation and default configuration, including the 5 sampling steps used in that codebase. The numbers in Table 6 come from the codebase's own evaluation path, not from a re-implementation. Its scores are dominated by a pronounced global color shift rather than residual text: PSNR, SSIM, and LPIPS are full-reference measures, so a shift applied across the frame penalizes every pixel, including the unedited background. These numbers therefore reflect color fidelity at least as much as subtitle-removal quality. We report them for completeness rather than as a like-for-like quality comparison. Two further behaviors matter for the comparisons in the main paper: CLEAR does not remove large-font subtitles, and its visual quality degrades as the sequence grows. For this reason, we assess long-video capability with the usability rate in Section 4.5 rather than with per-frame fidelity, because on minute-scale content per-frame metrics no longer show whether the result is usable.

F Proof of the One-Step Error Identity

This section proves the conditional-mean characterization and orthogonal risk decomposition used in Section 3.6, while retaining a general starting time t_0. The unconditional ideal-field risk is an expected conditional variance. No assumption about transport-path curvature is needed. We then discuss the low-uncertainty hypothesis and distinguish it from the available empirical evidence.

F.1 Setup and notation

Let $\mathbf{x}_0$ be a subtitle-free video and $\mathbf{x}_{\text{ref}}$ the corresponding video with subtitles rendered on it. Both are encoded by the frozen VAE encoder $\mathcal{E}$, giving $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ and $\mathbf{z}_{\text{ref}} = \mathcal{E}(\mathbf{x}_{\text{ref}})$. Let $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be drawn independently of the pair $(\mathbf{z}_0, \mathbf{z}_{\text{ref}})$, and let

$$\mathbf{z}_t = (1-t)\,\mathbf{z}_0 + t\,\epsilon, \qquad t \in [0,1], \tag{7}$$

be the rectified-flow interpolation of Equation 1, so that $t = 0$ is the clean latent and $t = 1$ is pure noise. Write $R = \mathbf{z}_{\text{ref}}$ as in Section 3.6, and suppress the fixed text prompt. The deployed one-step setting starts from pure noise; for the general t_0 result, $\mathbf{z}_{t_0}$ is drawn from this interpolation jointly with $(\mathbf{z}_0, R)$, not from a learned numerical trajectory.

The regression target in Equation 1 is $\mathbf{v}_t = \epsilon - \mathbf{z}_0$. At each fixed t, define the exact field as the conditional expectation, which minimizes the unweighted squared regression loss at that time:

$$v^*(\mathbf{z}_t, t \mid \mathbf{z}_{\text{ref}}) \;=\; \mathbb{E}[\epsilon - \mathbf{z}_0 \mid \mathbf{z}_t,\, \mathbf{z}_{\text{ref}}]\,. \tag{8}$$

We write $\mathrm{Cov}(\cdot \mid \cdot)$ for a conditional covariance matrix and Tr for its trace. The quantity $\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid R)$ depends on R; its outer expectation is the unconditional ideal endpoint risk. Unless conditioned or subscripted, expectations are over the evaluation law of $(\mathbf{z}_0, R, \epsilon)$, and $\mathbb{E}\|\mathbf{z}_0\|^2 < \infty$ is assumed throughout. For learned-field statements, training is complete and the measurable network v_θ is held fixed, with square-integrable field error at the time under study. Identities between random variables hold almost surely. Defining v^* at $t = 1$ explicitly is essential: a time-averaged loss with no endpoint mass does not determine the field at that single time.

F.2 One-step Euler sampling returns a conditional mean

Proposition 2. *Fix $t_0 \in (0, 1]$ and define the one-step Euler estimate obtained by integrating the exact field equation 8 from t_0 down to 0, i.e. with step size $\Delta t = -t_0$:*

$$\hat{\mathbf{z}}_0 \;=\; \mathbf{z}_{t_0} - t_0\, v^*(\mathbf{z}_{t_0}, t_0 \mid \mathbf{z}_{ref}). \tag{9}$$

Then

(i) $\hat{\mathbf{z}}_0 = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{ref}]$;

(ii) $\mathbb{E}\big\|\hat{\mathbf{z}}_0 - \mathbf{z}_0\big\|^2 = \mathbb{E}[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{ref})] \le \mathbb{E}_{\mathbf{z}_{ref}}[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{ref})]$;

(iii) at $t_0 = 1$ the inequality in (ii) holds with equality, and $\hat{\mathbf{z}}_0 = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{ref}]$.

Proof. (i) From equation 7 we have the algebraic identity

$$\mathbf{z}_{t_0} - t_0\,(\epsilon - \mathbf{z}_0) = (1 - t_0)\mathbf{z}_0 + t_0\epsilon - t_0\epsilon + t_0\mathbf{z}_0 = \mathbf{z}_0, \tag{10}$$

which holds pathwise: knowing the true increment $\epsilon - \mathbf{z}_0$ makes one Euler step exact. Taking $\mathbb{E}[\,\cdot \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}}]$ on both sides of equation 10, and using that $\mathbf{z}_{t_0}$ is measurable with respect to the conditioning variables and the definition equation 8, gives

$$\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}}] = \mathbf{z}_{t_0} - t_0\,\mathbb{E}[\epsilon - \mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}}] = \mathbf{z}_{t_0} - t_0\, v^*(\mathbf{z}_{t_0}, t_0 \mid \mathbf{z}_{\mathrm{ref}}) = \hat{\mathbf{z}}_0. \tag{11}$$

(ii) By (i), $\hat{\mathbf{z}}_0$ is the L^2 projection of $\mathbf{z}_0$ onto the σ-algebra generated by $(\mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}})$. Thus, its mean squared error is the expected conditional variance, $\mathbb{E}\|\hat{\mathbf{z}}_0 - \mathbf{z}_0\|^2 = \mathbb{E}[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}})]$. Applying the law of total variance conditionally on $\mathbf{z}_{\mathrm{ref}}$,

$$\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{\mathrm{ref}}) = \mathbb{E}[\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}}) \mid \mathbf{z}_{\mathrm{ref}}] + \mathrm{Cov}(\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{t_0}, \mathbf{z}_{\mathrm{ref}}] \mid \mathbf{z}_{\mathrm{ref}})\,, \tag{12}$$

Both terms on the right are positive semi-definite. Taking traces and expectations gives the stated inequality.

(iii) At $t_0 = 1$, Equation equation 7 gives $\mathbf{z}_1 = \epsilon$, which is independent of $(\mathbf{z}_0, \mathbf{z}_{\mathrm{ref}})$. Hence conditioning on $\mathbf{z}_1$ adds no information about $\mathbf{z}_0$: $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{z}_{\mathrm{ref}}] = \mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{\mathrm{ref}}]$ and $\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_1, \mathbf{z}_{\mathrm{ref}}) = \mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{\mathrm{ref}})$, so the second term in the law of total variance vanishes. □

Part (iii) proves Proposition 1: the unconditional ideal endpoint risk is $\mathbb{E}_R[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid R)]$, and it vanishes if and only if $\mathbf{z}_0$ is an almost-sure function of R. This is Bayes optimality for latent squared error, not conditional-distribution sampling or a perceptual guarantee after VAE decoding. Part (ii) compares the ideal risk at an interpolated starting point with the pure-noise risk; it is not a claim about intermediate states on a learned sampling trajectory. We do not use partial noising in the reported one-step setting.

F.3 The effect of the learned velocity field

Proposition 3. *Fix $t_0 \in (0, 1]$ and a trained measurable field v_θ. Let $\delta_{t_0} = v_\theta(\mathbf{z}_{t_0}, t_0 \mid R) - v^*(\mathbf{z}_{t_0}, t_0 \mid R)$ and assume $\mathbb{E}\|\delta_{t_0}\|^2 < \infty$ under the interpolation law. For $\hat{\mathbf{z}}_0^\theta = \mathbf{z}_{t_0} - t_0 v_\theta(\mathbf{z}_{t_0}, t_0 \mid R)$,*

$$\mathbb{E}\big\|\hat{\mathbf{z}}_0^\theta - \mathbf{z}_0\big\|^2 = \mathbb{E}[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{t_0}, R)] + t_0^2\,\mathbb{E}\|\delta_{t_0}\|^2. \tag{13}$$

At $t_0 = 1$, this is Equation 4.

Proof. Let $\mathcal{G} = \sigma(\mathbf{z}_{t_0}, R)$ and $m = \mathbb{E}[\mathbf{z}_0 \mid \mathcal{G}] = \hat{\mathbf{z}}_0$ by Proposition 2. Then $\hat{\mathbf{z}}_0^\theta - \mathbf{z}_0 = (m - \mathbf{z}_0) - t_0\delta_{t_0}$. The fixed-field error δ_{t_0} is $\mathcal{G}$-measurable, and square integrability makes the following cross term well-defined:

$$\mathbb{E}\langle m - \mathbf{z}_0, \delta_{t_0}\rangle = \mathbb{E}\langle \mathbb{E}[m - \mathbf{z}_0 \mid \mathcal{G}], \delta_{t_0}\rangle = 0. \tag{14}$$

Expanding the squared norm and using $\mathbb{E}\|m - \mathbf{z}_0\|^2 = \mathbb{E}[\mathrm{Tr}\,\mathrm{Cov}(\mathbf{z}_0 \mid \mathbf{z}_{t_0}, R)]$ proves the exact decomposition. Independence at $t_0 = 1$ gives the endpoint identity. □

The decomposition holds for any fixed field satisfying the integrability assumption, regardless of how it was trained. Thus subtitle-region weighting in Equation 2 does not invalidate it, but it neither identifies the weighted objective's optimizer with v^* nor guarantees a small unweighted endpoint error. For the unweighted objective, excess population regression risk at a fixed time equals $\mathbb{E}\|v_\theta - v^*\|^2$ by the same projection argument. A loss averaged over sampled times, however, does not automatically control this error at $t = 1$: when the endpoint has zero probability, changing the field there leaves that average unchanged. An endpoint bound needs additional endpoint supervision or quantitative assumptions that link nearby times to the endpoint, together with control under the evaluation distribution.

F.4 Why low conditional uncertainty is plausible for subtitle erasure

Let $\mathbf{M}_{\text{latent}}$ denote the set of latent coordinates whose VAE receptive field intersects the spatio-temporal support of the rendered subtitles; because $\mathcal{E}$ compresses space and time, $\mathbf{M}_{\text{latent}}$ is a dilation of the subtitle region rather than its exact image, and we do not assume the two coincide.

Outside $\mathbf{M}_{\text{latent}}$, the pixels of $\mathbf{x}_0$ and $\mathbf{x}_{\text{ref}}$ are identical by construction of the rendering pipeline, and $\mathcal{E}$ is deterministic. This suggests that the corresponding coordinates of $\mathbf{z}_0$ are strongly constrained by $\mathbf{z}_{\text{ref}}$, subject to effects from the VAE receptive field and reconstruction. This is consistent with the copy-preservation behavior reported in Section 4; it follows from the paired construction rather than from a separately implemented mechanism.

Inside $\mathbf{M}_{\text{latent}}$, adjacent frames may reveal occluded content as subtitles change or the camera moves, while surrounding pixels constrain texture and illumination (Appendix G). These cues motivate the hypothesis that residual conditional uncertainty is often localized and small, but do not determine its magnitude. Proposition 2(iii) characterizes ideal reconstruction risk; accuracy of the learned one-step output additionally requires a small endpoint field error.

The same reasoning suggests limits to the hypothesis: persistent occlusion may leave several plausible completions. Their latent conditional mean can be perceptually unrepresentative, although its decoded appearance is not determined by the latent risk identity. This is a possible ambiguity mechanism, not an established explanation of the reported failures. Likewise, multimodality in stylization would distinguish mean reconstruction from distributional sampling, but does not by itself imply that multiple steps are necessary. The 15-step stylization setting in Appendix G is an empirical configuration, not a lower bound on the required step count.

F.5 Empirical consistency and unmeasured uncertainty

The identity separates the expected conditional covariance trace from endpoint field error; neither term has been directly measured here. The 1-, 2- and 4-step results agree to within 0.30 dB PSNR (Table 1), and 30-step sampling is worse than 1-step (Table 6). These observations are consistent with effective one-step reconstruction but do not causally establish low conditional uncertainty or explain multi-step degradation. The 15-step stylization configuration (Appendix G) is not a controlled test of step necessity.

Seed sensitivity of the one-step estimator. At $t_0 = 1$ the starting point is pure noise and the exact-field one-step estimate is $\mathbb{E}[\mathbf{z}_0 \mid \mathbf{z}_{\text{ref}}]$, which does not depend on ϵ. We measure seed sensitivity of the decoded output as an empirical stability diagnostic, not as a bound on endpoint field error. We run the base model at one step with 8 seeds on all 400 VSR-Bench-400 clips, take the per-pixel sample variance across seeds (unbiased, averaged over RGB), and average it over the region pixels of a clip to obtain one value per clip. The median clip value is 6.0×10^{-5} inside the subtitle band and 4.7×10^{-5} outside it on a $[0, 1]$ intensity scale, i.e. a seed-induced standard deviation of roughly $2/255$. The band is modestly the noisier region – median ratio 1.15, mean ratio 1.37, and 249 of 400 clips with in-band above out-of-band – with the mean pulled up by a minority of

clips with large or heavily textured subtitles, the worst reaching 2.5×10^{-3} (about $13/255$). These measurements indicate low seed sensitivity for most clips, with variation present both inside and outside the subtitle band. Low seed variance does not establish proximity to the conditional-mean estimator: a seed-independent prediction can still have substantial bias. This measurement does not estimate the conditional covariance itself: at $t_0 = 1$ a single step cannot expose it, and a multi-step probe would instead estimate sampler-induced variability, which can be small even when the target distribution is uncertain if the model has collapsed.

G Additional Analyses and Discussions

G.1 Evaluation protocol details

Deriving the GSB judgment. Each VSR-Bench-400 clip in the study of Section 4.3 is seen by a single rater, who scores both outputs on the same 5-point scale. We therefore report a paired preference instead of two absolute means: an absolute score mixes output quality with that rater's personal use of the scale, whereas comparing the two scores given by the same rater accounts for their calibration. G means SEDiT scores higher, B means the baseline scores higher, and S means the two scores are equal, which covers both the case where neither output leaves a visible residual and the case where the residuals are equally severe.

Residual screening. The 30 real videos are horizontal 1920×1080 and vertical 1080×1920 clips. For the counts in Table 2, frames are sampled before being passed to the detector: the real videos at 1 fps, and each VSR-Bench-400 clip by 8 frames spread over the clip. The automatic pass is therefore a screening step rather than exhaustive per-frame detection, which is why every reported count was additionally verified by frame-by-frame human inspection of the full clip. Residuals are counted at the video level: a video counts as a residual if any frame of the full clip still shows subtitle content. Under the usability criterion of Section 4.5, RTF is defined as wall-clock time divided by video duration, so lower is faster and RTF < 1 is faster than real time.

G.2 Held-out corner-case protocol

We plan to expand the 30 real minute-scale videos of Section 4 to 50 source videos. Qwen-based mining will identify candidate videos with residuals in the base-model output, after which human reviewers will confirm the candidates. Selection will not use the distilled-model output and will exclude videos used for training, pseudo-label generation, hyperparameter tuning, checkpoint selection, or prompt development; duplicate or near-duplicate sources will also be removed. Because the set is selected from confirmed base-model failures, the base-model denominator is $50/50$ by construction and should not be interpreted as a population residual rate. We will report the number of residuals remaining after distillation and the recovery count out of 50, while separately recording background damage and temporal artifacts.

G.3 Ablation on subtitle-region loss weighting



Figure 12: Ablation study for subtitle-region weighting in Equation 2. Top row: "w/o subtitle-region weighting" ($\alpha = 0$, the unweighted flow-matching loss); bottom row: "w/ subtitle-region weighting" (our default, $\alpha = 5$). *Best viewed when zoomed in.*

G.4 END-TO-END COST BREAKDOWN

Of the end-to-end real-time factor (RTF) of 5.5 reported in Section 4.5, the model itself – VAE encoding, the single denoising step, and VAE decoding – accounts for an RTF of 1.5. The rest is shot detection plus video decoding and re-encoding. Decoding and re-encoding are needed by any pipeline that reads and writes video files, while shot detection is part of our own pipeline. Running inference only on the lower half of the frame, where film and television subtitles usually appear, lowers the end-to-end figure to 4.0. This variant cannot handle subtitles placed elsewhere, including the vertical layouts of Section 3.3, so it is an optional deployment setting rather than our default. All reported fidelity results use full-frame processing.

G.5 ON THE INTERPRETATION OF FULL-FRAME FIDELITY METRICS

Because the subtitle band covers only a small fraction of each frame, the four full-frame metrics of Table 1 are not equally informative about erasure quality. To quantify this, we evaluate under the same evaluation path the trivial output that returns the subtitled input unchanged and therefore performs no erasure at all. It obtains 24.93 PSNR, 0.9252 SSIM, 0.0932 LPIPS and 98.26 FVD.

PSNR and FVD behave as intended: they place this output below every method in Table 1, and measured against it SEDiT (1-step) recovers 83.1% of the available PSNR range and 87.3% of the available FVD range up to the VAE reconstruction reference. SSIM and LPIPS do not: the unchanged input scores higher than every method on both, and its SSIM even exceeds the VAE round-trip reference, which is possible only because it never passes through the VAE. The mechanism is that both measures average over local windows or patches, so a small region that is entirely wrong costs less than a global change of comparable magnitude introduced by re-encoding.

This does not invalidate the comparisons in Table 1, where every method is measured identically, but it does mean that small full-frame SSIM or LPIPS differences on this benchmark should not be read as differences in erasure quality. The region-resolved protocol of Table 1 addresses this directly by reporting fidelity inside the subtitle band, outside it, and on its boundary separately; these region-level results, together with PSNR, FVD, the human comparison, and the real-video demonstrations, provide complementary evidence for subtitle-erasure quality. Our quality claims rest on that breakdown, on PSNR and FVD, on the human side-by-side comparison of Section 4.3, and on the usability rate of Section 4.5.

G.6 STROKE-LEVEL DECOMPOSITION OF THE IN-BAND METRIC

The subtitle band of Table 1 contains two kinds of pixels. The *stroke* pixels are those the subtitle rendering actually changed: a pixel inside the filled subtitle box is a stroke pixel when the maximum absolute difference across the three RGB channels between the subtitled input and the clean target exceeds 20 intensity levels on the 0–255 scale. The comparison is per channel, not on a grayscale conversion and not on the channel mean, so a difference in any single channel is sufficient; glyph bodies, outlines and shadows are therefore all included, and every method has to synthesize content there. Pooled over the benchmark, stroke pixels are 30.9% of the box and 2.4% of the frame, and the box itself is 7.8% of the frame; the per-clip stroke share has median 28.8% and a 10th–90th percentile range of 15.1%–48.0%. The remaining pixels of the filled box, *band*$\setminus$*stroke*, are identical in the input and in the target and need no modification at all. This split uses the paired ground truth and is never supplied to any method; the mask-based baselines receive the filled box exactly as in Table 1. One clip contains no pixel meeting the stroke criterion, so stroke statistics are over 399 clips and the non-stroke and in-band statistics over all 400.

Table 7 reports the split. Two observations follow. First, the gap appears in both pixel classes, not only where the box protocol forces regeneration: SEDiT is 6.96 dB above the strongest baseline on the stroke pixels and 8.74 dB above it on the remainder of the band. This tells us where the gap sits, but not what causes it, because in both classes the baselines' input is zeroed over the whole box; we return to the input side below. Second, the cost of a box mask shows up exactly where one would expect. The baselines reach only 21.73–22.26 dB on pixels that were already correct in their own input, between 0.8 and 1.2 dB above their stroke scores, so the box protocol damages the unoccluded part of the band almost as much as the occluded part. SEDiT keeps 31.01 dB there, 2.85 dB above its own stroke level. The VAE round-trip reference shows how much of the remaining gap comes

from reconstruction rather than erasure: 4.80 dB of headroom on the stroke pixels against 2.51 dB on the rest of the band. This fits the localization-granularity bottleneck raised in the introduction: a bounding box forces an inpainting pipeline to regenerate the whole band, whereas a mask-free model only has to remove the glyphs. The four baselines cover GAN, diffusion-UNet and DiT backbones, use up to 50 sampling steps, and differ in cost by a factor of 28 per chunk. Even so, they fall within 0.7 dB of one another on the stroke pixels and within 0.6 dB on the rest of the band, which is consistent with the shared box-mask input, rather than the inpainting backbone, being the limiting factor.

The protocol is not biased in a single direction. Giving the baselines the filled box helps them on localization – they never pay for the OCR or tracking errors that a deployed pipeline would incur – and hurts them on granularity, since the box forces them to regenerate in-band pixels that were already correct. The decomposition above removes the metric side of this asymmetry by scoring stroke and non-stroke pixels separately, but it does not remove the input side: the baselines' input is zeroed over the entire box, so they lose the inter-character background as well, and with a stroke-level mask they would keep that context and could improve on the stroke pixels too. Changing the scoring region therefore cannot isolate the effect of mask granularity, and we make no such claim. We keep the box condition because a stroke-level mask is not available to a deployed mask-based pipeline without a dedicated text-matting or interactive segmentation stage, which is precisely the system complexity this work avoids. These numbers should therefore be read as the box-granularity condition – the condition under which such pipelines actually run – rather than as an upper bound on what those methods could reach with a finer mask.

Table 7: Stroke-level decomposition of the subtitle band on VSR-Bench-400, with MAE on a 0–255 scale. A box pixel counts as *stroke* when the maximum absolute difference across the three RGB channels between the subtitled input and the clean target exceeds 20 intensity levels; *band*$\backslash$*stroke* is the remainder of the filled box, which is identical in the input and the target. Stroke pixels are 30.9% of the box and 2.4% of the frame. Stroke columns are averaged over the 399 clips that contain such pixels, the non-stroke columns over all 400. All rows come from the same script over the same clips.

		stroke		**band$\backslash$stroke**	
Method	**Mask at inference**	PSNR ↑	MAE ↓	PSNR ↑	MAE ↓
EffectErase (50-step)	GT box	20.52	20.37	21.73	16.62
Propainter	GT box	20.91	19.05	22.06	15.69
DiffuEraser	GT box	21.20	17.97	22.26	15.27
Minimax-Remover (6-step)	GT box	21.00	18.98	21.79	16.65
SEDiT (1-step)	none	28.16	8.03	31.01	5.64
SEDiT-Distill (1-step)	none	28.03	8.11	30.86	5.72
VAE round-trip (reference)	–	32.96	4.77	33.51	4.36

G.7 Spatial-temporal cues of subtitles

Although our method does not use explicit masks, it localizes subtitles through spatial-temporal cues, which naturally distinguish subtitles from background content.

As summarized in Table 8, subtitles behave in a stable and structured way, while background regions change naturally and irregularly over time. Even when both change, the two patterns often differ, which may provide cues for locating subtitles without explicit supervision.

G.8 The Impact of User Prompt

In Figure 13, "Remove watermark" uses the prompt "Please remove the **watermark** from the video while preserving the character appearance, background composition, and color style. Do not add any new elements." "Remove subtitle" uses our predefined prompt, and "None" means an empty prompt (""). The results suggest that the wording of the prompt has little effect on the output, because training was set up as a single task. Since the prompt is fixed, we do not need to load the

Table 8: Subtitles and background follow distinct spatial-temporal patterns.

Case	Subtitle Variation	Background Variation	Spatial-Temporal Pattern	Subtitle Localization	Typical Scenario
1. Subtitle changes, background nearly static	✓ Significant changes (e.g., sentence switch, fade-in/out)	× Almost unchanged	Subtitle variation ≫ background variation	**Easy to localize**	Static camera, subtitles updating
2. Subtitle static, background changes	× Nearly unchanged	✓ Significant changes (camera motion, object movement, illumination changes)	Background variation ≫ subtitle variation	**Easy to localize**	Subtitle persists while the scene is dynamic
3. Subtitle static, background static (extreme case)	× Unchanged	× Unchanged	No spatial-temporal cues → relies purely on spatial cues	Difficult (must infer from subtitle style/shape)	Very short clips, fully static scenes
4. Subtitle and background both change	✓ Changes	✓ Changes	Different change patterns: subtitles are localized & structured; background changes are natural & irregular	Locatable (pattern-based)	Camera motion + subtitle transitions

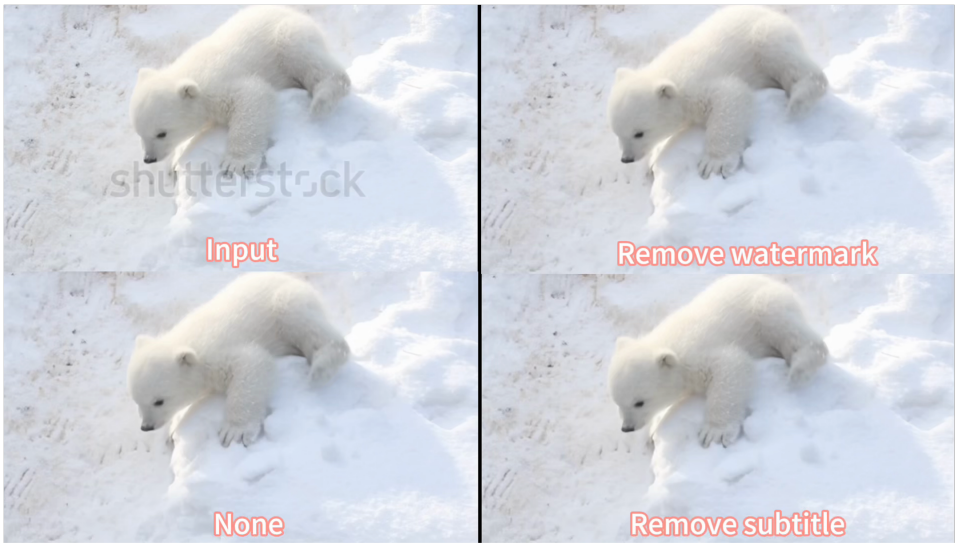


Figure 13: Results of watermark removal.

text encoder at inference: we use precomputed prompt embeddings instead, which lowers memory use and speeds up inference.

G.9 Optional Region-Constrained Editing

Sometimes a user wants to erase subtitles only inside a chosen region and keep the rest of the frame untouched. Our model supports this without retraining. Given a user-specified binary mask $\mathbf{M}_{\text{user}}$, we protect the region directly in VAE latent space. The output latent is $\mathbf{z}_{\text{out}} = \text{SEDiT}(\mathbf{z}_{\text{ref}}, \mathbf{z}_{\text{in}}) \cdot \mathbf{M}_{\text{latent}} + \mathbf{z}_{\text{ref}} \cdot (1 - \mathbf{M}_{\text{latent}})$, where $\mathbf{M}_{\text{latent}} = \text{Interpolate}(\mathbf{M}_{\text{user}}, (F_{latent}, H_{latent}, W_{latent}))$ is the user mask resized in space and time to the latent resolution. Outside the mask, the latent is copied from the reference, so the model's edit is confined to the selected region. Because the VAE mixes information over a spatial-temporal neighborhood, $\mathbf{M}_{\text{latent}}$ is coarser than the pixel mask the user draws – the same dilation effect discussed in Appendix F. Protection therefore holds at the level of latent coordinates: near the boundary, and after decoding, pixel values are still affected by the decoder's receptive field and by VAE reconstruction error. Figure 14 shows this mechanism applied to a logo area. The mask is entirely optional and is supplied by the user for creative control; subtitle removal itself does not need it, and the default pipeline stays mask-free at inference. In this sense, **SEDiT** also fits the region-constrained workflows of earlier mask-based models.



Figure 14: Example of Logo Protection. The blue box indicates the protected region. *Best viewed when zoomed in.*

G.10 Positional Encoding Selection

We use the same 3D RoPE design for the reference and noisy latents because video subtitle removal is aligned frame by frame: $\mathbf{u}_{\text{ref}} = \mathbf{u}_{\text{noisy}} = (f, h, w)$. In LTX-Video, f is at most 20, which corresponds to $20 \times 8 + 1 = 161$ frames. With a shifted scheme such as $\mathbf{u}_{\text{ref}} = (f, h, w)$ and $\mathbf{u}_{\text{noisy}} = (f + \delta, h, w)$, the largest usable f drops to 10, so the reference video is limited to $10 \times 8 + 1 = 81$ frames. A single chunk could then cover only 81 frames, which is not enough for our 121-frame setting at 720p. Another option is to add an extra axis that separates the two streams: $\mathbf{u}_{\text{ref}} = (0, f, h, w)$ and $\mathbf{u}_{\text{noisy}} = (1, f, h, w)$. This changes the coordinate scheme used by the pretrained 3D RoPE. We keep the original 3D scheme to preserve that structure and to keep matching reference and noisy tokens aligned.

G.11 The Visual Results for Video Style Transfer

To test how well the SEDiT architecture generalizes and how much the user prompt matters, we train a video stylization model on the recent video editing dataset Ditto-1M (Bai et al., 2025). As shown in Figure 15, the **user prompt** steers the model toward clearly different visual styles, which

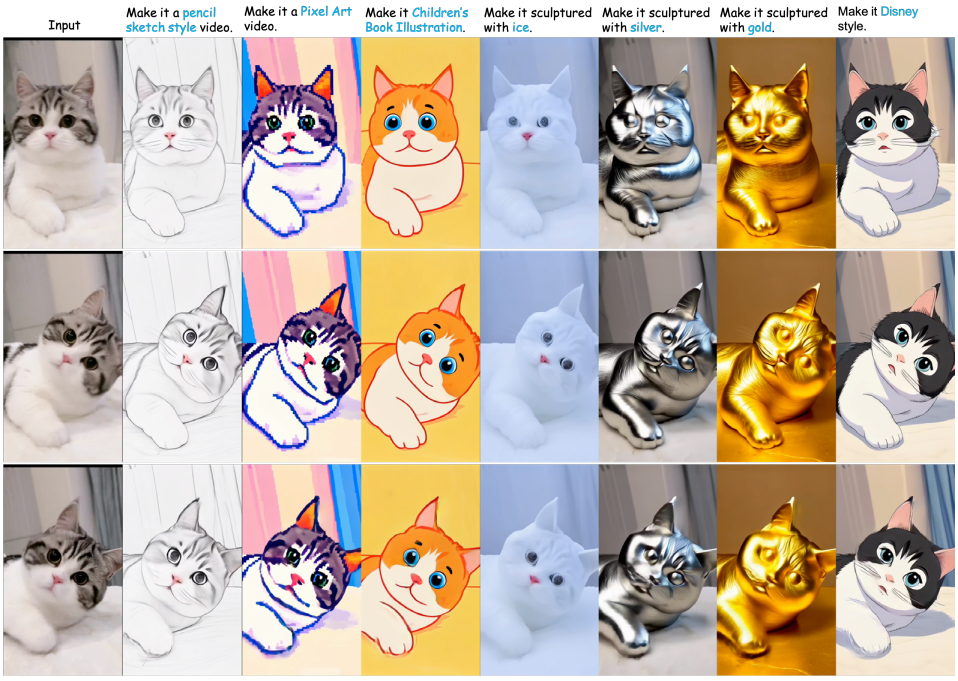


Figure 15: Visual Results for video style transfer.

suggests that the architecture can be adapted to other video editing tasks. Unlike subtitle removal, stylization changes the whole frame; the reported configuration uses 15 sampling steps, and we do not claim that this number of steps is necessary.

G.12 Failure case of OCR

As shown in Figure 16, we run a recent OCR model, PaddleOCR-VL (Cui et al., 2025), on videos that contain text in the scene. Besides the subtitles, it also picks up this other text. In the second row, first column, PaddleOCR-VL fails to read subtitles that use a transition effect. Detection errors like these can hurt a pipeline that depends on masks. Our method removes these subtitles while keeping the text that belongs to the scene.

H Supplementary Video Demonstrations

We provide supplementary video demonstrations as qualitative evidence complementary to the paired evaluation on VSR-Bench-400. Most examples are real videos rather than clips from VSR-Bench-400. They allow visual inspection of subtitle residuals, temporal artifacts, and changes to non-subtitle content that may be difficult to assess from selected still frames. The 30-video usability study in Section 4.5 primarily concerns television dramas and short-form dramas; its reported $25/30$ and $29/30$ results should be interpreted within that application scope rather than as a claim of exhaustive real-world coverage.



Figure 16: The OCR failure case in subtitle recognition. *Best viewed when zoomed in.*