DEOVERLAY: A BENCHMARK AND SOURCE-ANCHORED FLOW FOR VIDEO OVERLAY REMOVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Video overlay removal aims to recover scenes obscured by subtitles, titles, and logos while preserving surrounding content. Existing methods use temporal propagation or source-conditioned generation, but can leave residual overlays and distort scene structure when useful source information is obscured. We propose **DeOverlay**, a source-anchored flow model that reconstructs the clean video from the source latent with fixed source guidance. Specifically, we train a shared velocity network along paired source-to-clean paths using region-weighted flow matching, enabling successive corrections from intermediate states. To select local source cues, we develop visibility-normalized temporal aggregation that excludes masked positions and averages the remaining observations. Global source context and edit-region embeddings complement these cues through additive conditioning, retaining a single video-token sequence. To evaluate scene recovery across diverse overlay conditions, we introduce **DVTE-Bench**, comprising 1,631 paired videos across eight text-overlay categories, with OCR-screened clean targets, frame-aligned pixel masks, and a common evaluation protocol. Experiments on the complete release show that DeOverlay outperforms four baselines on all five reconstruction and perceptual metrics, reducing Mask LPIPS and Mask VFID by 58.3% and 70.3%, respectively, relative to the strongest evaluated baseline. Additional qualitative examples demonstrate the removal of graphic logos and stickers. We will make the complete benchmark, evaluation code, and test outputs available.

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.

1 INTRODUCTION

Video overlay removal aims to recover scene content obscured by subtitles, titles, watermarks, and logos while leaving the surrounding video unchanged. Although neighboring frames may reveal some occluded content, persistent coverage leaves other regions unobserved throughout the clip. Effective removal therefore requires both exploiting available observations and inferring missing content, while maintaining temporal coherence and preserving the scene structure already present in the input.

Propagation-based inpainting transfers image content and features across frames, but relies on learned priors when covered regions lack direct observations (Zhou et al., 2023). Generative methods infer covered content through subtitle-specific reconstruction (He et al., 2026) or noise-initialized editing. Noise-initialized editors incorporate source information through channel concatenation (Zi et al., 2025) or sequence concatenation (Hui & Bai, 2026), while the editable state begins from noise. Our paired comparisons show residual lettering and boundary distortions in some evaluated outputs (Figure 4).

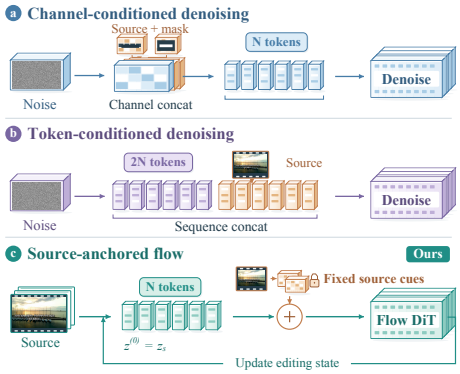

Figure 1: **Starting state and source conditioning.** Channel- and sequence-conditioned editors start from noise; ours starts from z_s and receives fixed source features at each update. On a shared patch grid, the three designs use N, $2N$, and N video tokens per call. The diagram omits auxiliary tokens and codecs.

We propose **DeOverlay**, which starts reconstruction from the source latent and keeps the original source available to each update. A shared velocity network learns source-to-clean corrections from paired latent paths under a region-weighted flow objective. At inference, the state and time evolve while the weights and source guidance remain fixed.

Visibility-normalized temporal aggregation averages unmasked source features across nearby frames. A global source summary supplies scene context, and an edit-region embedding marks the covered region. These terms are added to the editing tokens at every update, keeping the source available without a second video-token sequence. Where no frame reveals the covered content, the video prior infers it.

Evaluating overlay removal requires checking whether the recovered content agrees with the underlying scene. Existing studies assess subtitle erasure (He et al., 2026) or visible watermark removal (Chen et al., 2025) in separate task-specific settings. We introduce **DVTE-Bench**, the Diverse Video Text Erasure Benchmark, with 1,631 paired videos covering eight text-overlay categories. The categories vary in overlay layout, persistence, and appearance, while OCR-screened clean targets and aligned pixel masks support common regional evaluation. Experiments on the complete release show improvements over four baselines on all five reconstruction and perceptual metrics. Component and step-count studies further examine the contribution of source guidance and repeated updates.

Our contributions are summarized as:

- We introduce **DVTE-Bench**, a paired benchmark covering eight text-overlay types with clean targets, aligned pixel masks, and common evaluation of regional reconstruction, perceptual quality, and temporal behavior.
- We propose **DeOverlay**, which learns a source-to-clean flow with fixed local and global guidance. Source initialization retains observed structure, while visibility-normalized aggregation selects local cues for reconstructing covered regions.
- We evaluate DeOverlay on all 1,631 released clips and obtain the best results among five methods on all five reconstruction and perceptual metrics. Ablations examine source guidance and flow step count, with separate measurements of inference cost.

2 Related Work

Video inpainting and overlay removal. ProPainter combines image and feature propagation with mask-guided sparse attention (Zhou et al., 2023). For subtitle removal, CLEAR learns subtitle representations followed by generative reconstruction and supports mask-free inference (He et al., 2026). DECO combines frame-wise priors with temporal modeling for visible watermark removal (Chen et al., 2025). Our method learns source-to-clean corrections with persistent source guidance.

Source-conditioned restoration. SEDiT concatenates fixed source-reference tokens with noise-initialized tokens for one-step subtitle erasure (Hui & Bai, 2026). DiffST combines input-derived initialization, motion-based frame aggregation, and video-prompt guidance for space-time super-resolution (Chen et al., 2026). We adapt this local/global decomposition with visibility-normalized aggregation at fixed coordinates and a pooled latent summary, guiding a source-initialized state through successive updates.

Flow-based editing. Flow matching and rectified flow learn vector fields along prescribed paths (Lipman et al., 2023; Liu et al., 2023). IR-Flow learns degraded-to-clean image transport (Fan et al., 2026), while PCFlow compares observation-initialized unconditional flow with noise-initialized conditional flow (Jo et al., 2026). FlowEdit performs inversion-free flow editing (Kulikov et al., 2025), and RFDM predicts residuals relative to preceding outputs for causal video editing (Salehi et al., 2026). Our paired video flow combines source initialization, fixed source guidance, and regional weighting within a pretrained video transformer (Ma et al., 2026; Robbyant, 2026).

Evaluation resources. CLEAR evaluates a 400-clip subtitle benchmark (He et al., 2026), while SEDiT introduces VSR-Bench-400 with clean videos and subtitle masks (Hui & Bai, 2026). DECO evaluates stationary and moving watermarks synthesized on YouTube-VOS and DAVIS (Chen et al., 2025).

Table 1: **Comparison of video text erasure benchmarks.** CLEAR (He et al., 2026) and VSR-Bench-400 (Hui & Bai, 2026) are compared with DVTE-Bench. The DVTE-Bench release checkmark denotes its planned public release.

Benchmark	Clips	Release	Ground Truth		Overlay Diversity					Evaluation	
			Clean	Mask	Types	Multilingual	Spatial	Temporal	Appearance	Regional	Temporal
CLEAR test set	400	✗	✓	✗	1	✗	✓	✗	✓	✗	✓
VSR-Bench-400	400	✗	✓	✓	1	✓	✓	✗	✓	✗	✗
DVTE-Bench (ours)	**1,631**	✓	✓	✓	**8**	✓	✓	✓	✓	✓	✓

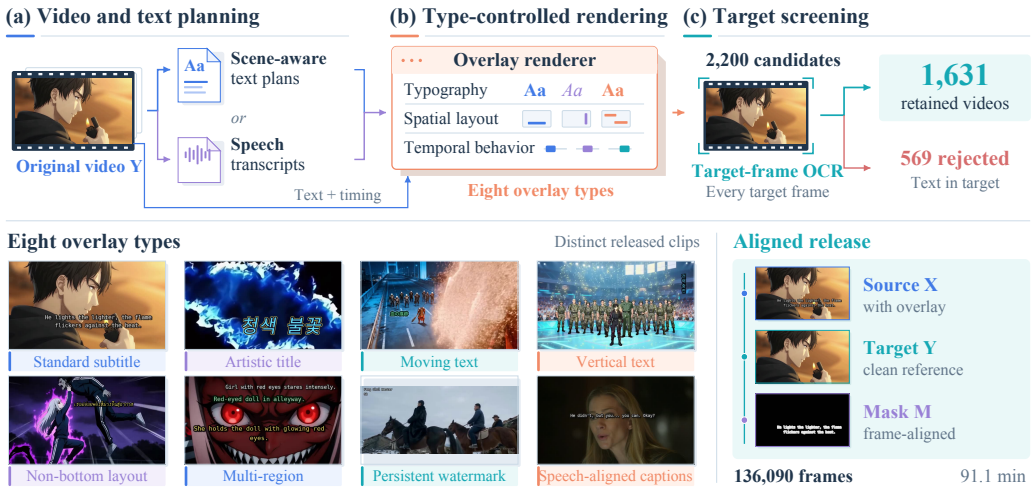

Figure 2: **DVTE-Bench construction.** Text plans and type-controlled rendering produce aligned source videos, clean targets, and overlay masks. Target-frame OCR screening rejects 569 of 2,200 candidates and retains 1,631 clips. The eight thumbnails show distinct released clips. The lower-right source, target, and mask form one aligned example.

3 DVTE-Bench: Diverse Video Text Erasure

We construct DVTE-Bench to evaluate overlay removal and scene reconstruction. Each sample pairs an overlaid video with a clean target and frame-aligned masks, allowing common evaluation across overlay layouts, temporal behaviors, and appearances.

Table 1 contrasts DVTE-Bench with existing subtitle benchmarks in paired supervision, overlay diversity, and evaluation coverage.

3.1 Challenge Design

The benchmark contains eight overlay types organized around three aspects of reconstruction (Table 7). *Spatial layout* determines which scene regions must be recovered, including vertical text, non-bottom placement, and multiple simultaneous regions. *Temporal behavior* affects whether nearby frames expose the covered content, as illustrated by moving text and persistent watermarks. *Appearance and coverage* vary through standard subtitles and artistic titles with display typography, outlines, or glow. Speech-aligned captions add audio-derived content and timing. These factors interact within each category, so category comparisons measure coverage across conditions rather than isolated effects.

3.2 Paired Data and Target Quality

The release contains 1,631 clips, 136,090 frames, and 91.1 minutes of video, with 1,085 portrait and 546 landscape clips. Figure 2 outlines construction. Scene-aware text plans or speech transcripts supply overlay content. Type-controlled rendering retains the original video as the reconstruction target and produces aligned overlay masks, including masks that follow moving text. Language

metadata has 11 labels, dominated by Chinese, English, Korean, and Japanese. Figure 6 shows the released distributions.

Text in a reference can be mistaken for an unerased overlay. We therefore screen every target frame with OCR and reject candidates with confirmed text, removing 569 of 2,200 candidates and retaining 1,631 (74.1%). Preservation of natural scene text requires a separate evaluation setting.

3.3 Regional Evaluation Protocol

All five methods are evaluated on the same 1,631 released clips. Scoring uses pixel-accurate GT masks from the released NPZ archives, independently of the bounding-box masks supplied to mask-conditioned models.

Mask PSNR measures pixel fidelity, and Mask SSIM (Wang et al., 2004) averages the SSIM map over GT mask pixels. Mask LPIPS (Zhang et al., 2018), DISTS (Ding et al., 2022), and VFID measure perceptual agreement with clean targets. For these perceptual metrics, exterior prediction pixels are replaced by the target, and a shared crop covers the union of mask locations over time. This construction focuses evaluation on the overlay region. Frame measurements are averaged within videos and then equally across videos. VFID is a global distance between video-feature distributions. Infinite-PSNR videos are omitted from the finite PSNR mean and counted as N_∞.

Temporal evaluation reports mask-restricted TWE for predictions and clean targets, their aggregate gap, and auxiliary TC and flow statistics using the released Farneback protocol. These diagnostics complement target-referenced fidelity because temporally smooth outputs can still reconstruct the wrong content. Appendix C describes the regional computations and aggregation.

4 DeOverlay: Source-Anchored Flow

Given an overlaid video, a removal instruction, and an edit mask, we first encode the source as the initial editing state. A shared velocity network then predicts corrections using fixed local and global source guidance. Finally, a video decoder maps the updated latent to the reconstructed video (Figure 3).

4.1 Source-Initialized Continuous Flow

To learn corrections from an overlaid video to its clean counterpart, we connect their latent representations with a paired flow path (Lipman et al., 2023; Liu et al., 2023). Let $X, Y \in \mathbb{R}^{T\times H\times W\times 3}$ denote the aligned source and target, with M marking affected pixels, including partially transparent regions. A video VAE encodes $z_s = \mathcal{E}(X)$ and $z_c = \mathcal{E}(Y)$ in $\mathbb{R}^{C\times T_\ell\times H_\ell\times W_\ell}$. Let $\mathcal{D}$ denote decoding and $\mathbf{c}$ the removal instruction. The training path is

$$a = z_s + \sigma\epsilon, \qquad z_\tau = (1-\tau)z_c + \tau a, \qquad \epsilon \sim \mathcal{N}(0, I), \quad \tau \in [0, 1], \tag{1}$$

with $\sigma = 0.05$. The source endpoint is at $\tau = 1$ and the clean endpoint at $\tau = 0$, giving the target velocity

$$v^* = \partial z_\tau / \partial \tau = a - z_c. \tag{2}$$

We sample intermediate τ so that the predictor $v_\theta(z_\tau, \tau, \mathbf{c}; z_s, M_{\text{model}})$ learns corrections at different positions along the path. It receives the current state and time together with the original source and conditioning mask. The perturbation augments training, while inference starts directly at z_s.

4.2 Persistent Source Guidance

As editing changes the reconstruction, fixed source guidance keeps the original observations available. We combine local source features, a global summary, and the edit mask with the patchified state by addition:

$$h_\tau = P(z_\tau) + A(z_s, M_{\text{model}}) + G(z_s) + m_{\text{tok}} e_M. \tag{3}$$

Here P embeds the state, m_{tok} is the token-grid mask, and e_M is a learned vector. All terms have the transformer width; G is broadcast over tokens. Addition preserves the number of video tokens.

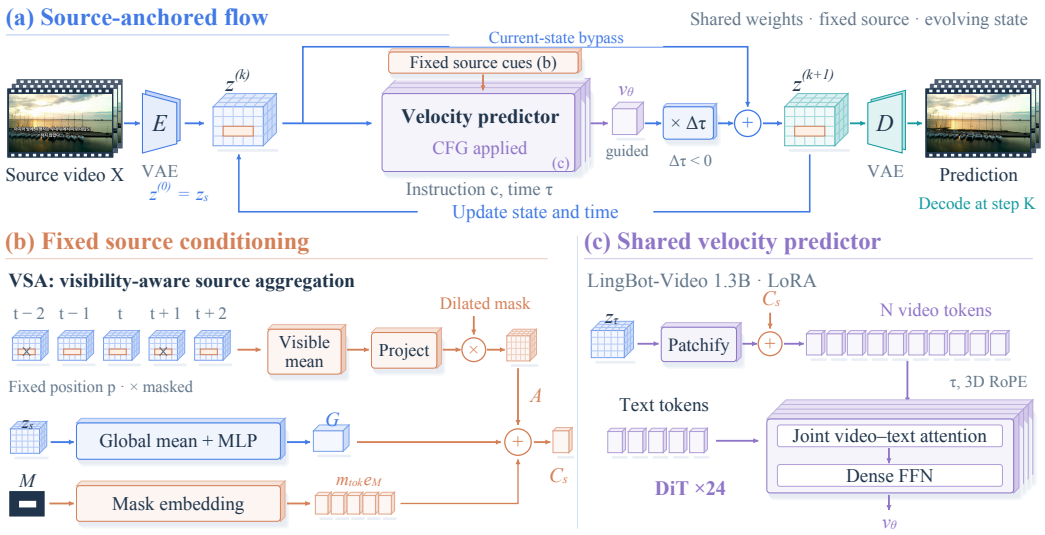

Figure 3: **DeOverlay overview.** (a) Shared-weight Euler updates start from the source latent. The upper bypass carries the current state; the lower loop updates state and time with fixed source cues. (b) Visibility-aware source aggregation (VSA), global source conditioning (GSC), and mask embeddings form the source condition. (c) Source cues are added to the current state tokens before joint video–text processing. Tensor grids and intermediate states are schematic; RGB frames are matched source/output pairs.

Visibility-aware source aggregation. Direct temporal averaging can mix overlay features with useful scene information. Inspired by the local/global decomposition in DiffST (Chen et al., 2026), our visibility-aware source aggregation (VSA) excludes masked positions and normalizes by the number of visible observations. Let $s_{t,p}$ be the source latent average-pooled to the token grid and $m_{t,p}$ its max-pooled mask. Within $\mathcal{N}(t) = \{t-2, \dots, t+2\}$, we compute

$$\bar{s}_{t,p} = \frac{\frac{1}{5}\sum_{j\in\mathcal{N}(t)}(1-m_{j,p})s_{j,p}}{\max\left(\frac{1}{5}\sum_{j\in\mathcal{N}(t)}(1-m_{j,p}), \delta\right)}, \qquad \delta = 10^{-4}. \tag{4}$$

The denominator prevents masked positions from attenuating the visible-feature mean. We project $\bar{s}$ to transformer width with a two-layer pointwise network and gate it to the edit region with a dilated mask. Aggregation uses fixed coordinates, so motion can mix structures and fully masked windows provide no local observations. Video attention supplies broader context.

Global source conditioning. Local observations may be unavailable under persistent coverage. Our global source conditioning (GSC) projects the source's spatiotemporal mean through a two-layer MLP to provide scene-level context, including both overlaid and uncovered positions. The resulting summary G complements local features, while $m_{\text{tok}} e_M$ marks the edit region. Output projections and e_M start at zero. Appendix A gives implementation details.

4.3 Learning the Velocity Field

To emphasize overlay correction, we assign larger loss weights to covered regions. Let $\widetilde{M}$ be the annotated mask resized to the latent grid and dilated by one cell. With q_i excluding overlap-conditioning elements, we minimize

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\frac{\sum_i q_i(1+\lambda\widetilde{M}_i)(v_{\theta,i}-v_i^*)^2}{\max(\sum_i q_i, 1)}\right], \qquad \lambda = 5. \tag{5}$$

The index spans channel, time, space, and batch. Covered elements receive weight six and exterior elements weight one; the denominator counts supervised elements without weighting.

We adapt the dense 1.3B LingBot-Video backbone (Ma et al., 2026; Robbyant, 2026) with rank-256 LoRA (Hu et al., 2022), directly training the patch embedding and source-guidance modules while freezing the VAE and text encoder. Conditioned video tokens and projected instruction tokens undergo joint attention. With probability 0.2, missing-mask training sets $M_{\text{model}} = 0$, zeroing local guidance and the edit embedding while retaining global guidance. Loss weighting still uses the annotated mask. Table 6 records the full configuration.

4.4 Shared-Weight Flow Updates

Starting at $z^{(0)} = z_s$, we use a descending schedule $1 = \tau_0 > \cdots > \tau_K = 0$:

$$z^{(k+1)} = z^{(k)} + \Delta\tau_k \, \widehat{v}_\theta(z^{(k)}, \tau_k, \mathbf{c}; z_s, M_{\text{model}}), \qquad \Delta\tau_k = \tau_{k+1} - \tau_k < 0. \tag{6}$$

The state and time change while weights, source, and mask remain fixed, so one checkpoint supports multiple step counts. The linear–quadratic schedule uses a resolution-dependent shift. Classifier-free guidance (Ho & Salimans, 2022) at scale $g = 2$ uses two network evaluations per step: six steps require 12 evaluations. The core output is $\widehat{Y} = \mathcal{D}(z^{(K)})$; chunked inference and output processing are specified in Appendix A.

Although each training pair defines a straight latent path, the learned velocity can vary with the state and time during inference. We therefore evaluate how the number of flow updates affects reconstruction quality in Section 5.3. Appendix B analyzes the integration and endpoint errors.

5 Experiments

5.1 Experimental Setup

Training, inference, and baselines. Training uses 350,000 aligned overlaid/clean clip pairs with masks: 250,000 base samples, 50,000 stylized-text samples, and 50,000 mirrored-text samples. The training set is disjoint from DVTE-Bench. Appendix A details the corpus and optimization settings. Full-release evaluation uses the 40K checkpoint, GT bounding-box input masks, $K = 6$, guidance scale $g = 2$ (12 network evaluations), and 12-pixel seamless paste-back. Baselines are ProPainter (Zhou et al., 2023), CLEAR (He et al., 2026), MiniMax-Remover (Zi et al., 2025), and DiffuEraser (Li et al., 2025), spanning propagation, subtitle-specific, and generative removal. Non-subtitle cases test CLEAR's transfer beyond its original task. All methods share GT scoring masks; input-mask access and postprocessing follow their released pipelines (Appendix D). SEDiT (Hui & Bai, 2026) is compared conceptually because an executable model was unavailable for our matched evaluation. Ablations use 200 randomly sampled videos from DVTE-Bench.

5.2 Reconstruction across Diverse Overlays

Table 2 compares reconstruction quality on all 1,631 released videos. DeOverlay achieves the best results on all five reconstruction and perceptual metrics. Compared with MiniMax-Remover, the strongest baseline on these metrics, it improves Mask PSNR from 26.17 to 30.80 dB and SSIM from 0.8044 to 0.8822. LPIPS, DISTS, and VFID decrease by 58.3%, 40.8%, and 70.3%, respectively, supporting more faithful covered-region reconstruction.

Coverage across overlay conditions. DeOverlay achieves the best PSNR, SSIM, LPIPS, and DISTS across all eight overlay types (Appendix D). Artistic titles produce its lowest PSNR/SSIM (26.24 dB/0.8409), whereas vertical text produces its highest LPIPS/DISTS (0.0975/0.1015). The most difficult category therefore depends on whether pixel accuracy or perceptual similarity is measured.

Temporal behavior. Table 2(b) reports the released temporal diagnostics alongside reconstruction fidelity. DeOverlay obtains TWE Pred of 9.3110 and TC of 4.2498, compared with ProPainter's 8.1428 and 4.0349. All methods use the same mask-restricted computation and clean-target reference of 9.3228. ProPainter and MiniMax-Remover score below this reference despite weaker reconstruction, illustrating why the diagnostic is interpreted together with clean-target fidelity. Appendix I specifies the temporal computations.

Table 2: **DVTE-Bench results on all 1,631 videos**, using the released pixel-mask protocol. (a) PSNR† excludes infinite values; VFID is a global video-feature distance. (b) TWE Gap is $|\overline{\mathrm{TWE}}_{\mathrm{pred}} - \overline{\mathrm{TWE}}_{\mathrm{GT}}|$, with clean-target TWE 9.3228. Diagnostics are defined in Appendix I. Bold marks best values in (a). In (b), bold/underline mark the closest/second-closest TWE Pred to GT and the lowest/second-lowest values otherwise.

(a) Regional reconstruction and perceptual quality

Method	PSNR† ↑	SSIM↑	LPIPS↓	DISTS↓	VFID↓
CLEAR	16.02	0.5172	0.3209	0.2158	1.4736
ProPainter	25.90	0.7832	0.1166	0.0957	0.3879
MiniMax-Remover	26.17	0.8044	0.0977	0.0856	0.2806
DiffuEraser	24.84	0.7723	0.1130	0.0952	0.3285
DeOverlay, $K = 6$	**30.80**	**0.8822**	**0.0408**	**0.0507**	**0.0835**

(b) Regional temporal diagnostics

Method	TWE Pred	TWE Gap	TC	Flow Mean	Flow Var
CLEAR	10.2373	0.9144	11.0261	0.9014	2.8169
ProPainter	8.1428	1.1801	**4.0349**	0.6496	1.5601
MiniMax-Remover	8.7095	0.6133	4.5586	0.7991	2.3072
DiffuEraser	10.1853	0.8624	5.3240	0.6690	1.9385
DeOverlay	**9.3110**	**0.0118**	4.2498	**0.3333**	**0.5921**

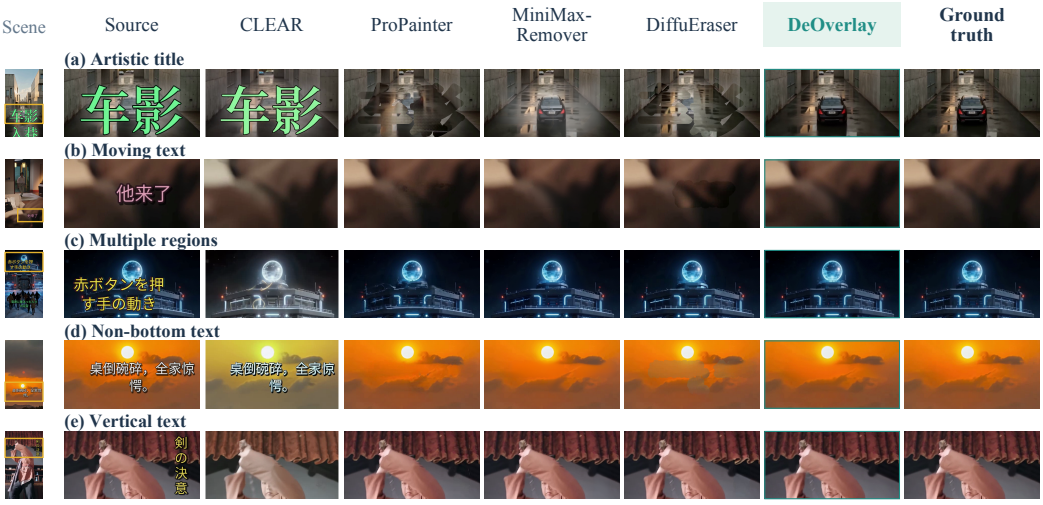

Figure 4: **Paired benchmark comparisons.** Five frame-40 examples with shared yellow crop boxes compare four baselines, DeOverlay, and clean targets. Appendix E gives inference settings and expanded seven-category comparisons.

DeOverlay has an aggregate TWE Gap of 0.0118, versus 0.6133 for MiniMax-Remover and 1.1801 for ProPainter. This gap compares dataset-level means under the aggregation rule in Appendix I. The five target-referenced metrics in Table 2(a) therefore provide the primary reconstruction comparison. Flow Mean, Flow Var, and TC retain their released labels and scales as supplementary descriptions of the evaluated outputs.

Qualitative reconstruction. Figure 4 compares five paired examples with clean targets. Residual text, blurred structures, and altered colors distinguish the outputs, while DeOverlay more closely preserves the target appearance. In the artistic-title example, the restored car and road remain aligned with the clean target. The multiple-region and non-bottom-text examples make color shifts and damaged background boundaries visible around the building and clouds. These details complement the aggregate scores by showing which structures are recovered within the covered regions. Figure 5 adds twelve application examples spanning subtitle languages and density, title layouts, watermarks, a sticker, and a television logo. Appendix E provides seven-category comparisons, application full frames, and additional successes and failures.

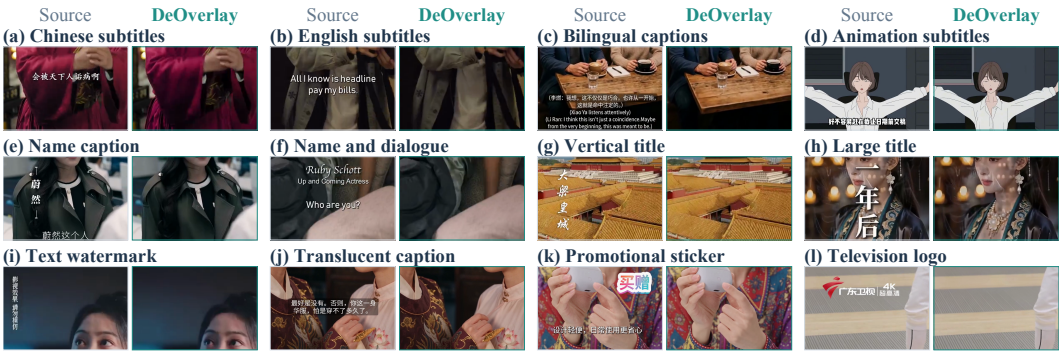

Figure 5: **Removal across overlay appearances.** Twelve matched source/output crops cover subtitles, titles, watermarks, a sticker, and a television logo. Full frames and crop boxes appear in Appendix E. These application clips have no clean targets.

Table 3: **Core design ablations on 200 randomly sampled videos.** Component variants use $K = 4$; the direct-residual reference uses $K = 1$. PSNR and MAE use overlay pixels. VSA supplies local source context and GSC supplies global context. The noise-start variant concatenates noise and source video tokens. ΔPSNR is relative to the full four-step model. Bold marks the best value in each quality column.

Variant	PSNR↑	MAE↓	Crop SSIM↑	Outside MAE↓	TWE gap↓	ΔPSNR
DeOverlay, $K = 4$	**32.461**	**4.299**	**0.92765**	**2.541**	**0.180**	0.000
Direct residual, $K = 1$	31.300	4.520	0.91120	2.548	0.195	-1.161
Noise init. + source tokens	29.600	6.500	0.89200	3.050	0.320	-2.861
Mask only (no VSA/GSC)	32.230	4.440	0.92590	2.552	0.197	-0.231
Ordinary temporal mean	32.360	4.360	0.92725	2.542	0.183	-0.101
Without VSA	32.370	4.350	0.92705	2.543	0.185	-0.091
Without GSC	32.290	4.410	0.92645	2.550	0.193	-0.171

Crop SSIM denotes MaskCrop SSIM; TWE gap denotes the TWE absolute gap.

5.3 Ablation Studies

Ablations use 200 randomly sampled DVTE-Bench videos; Table 2 uses all 1,631 videos. Component variants use $K = 4$, with a direct-residual reference at $K = 1$; the step sweep holds the checkpoint fixed. Ablation SSIM is measured with MaskCrop SSIM.

Effect of source representation and guidance. We first replace source initialization and additive guidance with noise initialization and concatenated source tokens (Table 3). Mask PSNR decreases by 2.861 dB. Because the starting state and conditioning change together, this comparison evaluates their combined design.

We then vary the guidance within the source-initialized model. Keeping only the mask embedding reduces PSNR by 0.231 dB, while removing VSA or GSC reduces it by 0.091 or 0.171 dB. Both local observations and global context therefore contribute to the measured reconstruction quality. Replacing visibility-normalized aggregation with an ordinary temporal mean reduces PSNR by 0.101 dB and raises MAE to 4.360, supporting the use of visibility selection in local aggregation.

Effect of flow step count. Increasing from two to four steps within the same checkpoint improves Mask PSNR from 32.410 to 32.461 dB, lowers MAE from 4.330 to 4.299, and improves Crop SSIM and TWE gap (Table 4). Eight steps add only 0.007 dB over four at $1.50\times$ latency, favoring four steps for these component studies.

Training formulation and repeated updates. Four-step flow exceeds the separately trained direct-residual reference by 1.161 dB in Mask PSNR, with lower MAE and higher Crop SSIM (Table 3). Each flow update recomputes the velocity from the current state and time with fixed source cues, allowing different inference budgets without retraining.

Table 4: **Flow step count on 200 randomly sampled videos.** The checkpoint is fixed; only the inference update count changes. Latency is normalized to $K = 4$. Bold marks the best quality metric.

K	Mask PSNR↑	Mask MAE↓	MaskCrop SSIM↑	TWE abs. gap↓	Rel. latency
2	32.410	4.330	0.92710	0.184	0.82×
4	32.461	4.299	0.92765	0.180	1.00×
6	32.467	4.294	0.92772	**0.178**	1.25×
8	**32.468**	**4.293**	**0.92774**	**0.178**	1.50×

Table 5: **Speed and memory on the same A100.** All rows process the same eight clips (650 frames). E2E denotes end-to-end time excluding model initialization. Inference time follows each runner's CUDA-synchronized interval. Peak memory uses a common external GPU-memory sampler. Bold marks the best value in each column.

Method	E2E s/frame↓	E2E FPS↑	Inference s/frame↓	Peak GiB↓
CLEAR	0.6030	1.6584	0.5983	18.354
ProPainter	0.4271	2.3413	0.4134	34.101
MiniMax-Remover	0.3787	2.6405	0.3530	**10.776**
DiffuEraser	1.1677	0.8564	1.1524	23.950
DeOverlay, $K = 1$	**0.3242**	**3.0847**	**0.3001**	12.236
DeOverlay, $K = 4$	0.4828	2.0712	0.4587	12.236

5.4 Efficiency

End-to-end speed. Table 5 uses eight shared clips (650 frames), run serially on one A100. Timing includes decoding, postprocessing, and encoding, excluding model loading. One-step DeOverlay reaches 3.0847 FPS, 1.17× MiniMax-Remover's throughput. Four steps reach 2.0712 FPS. Sampled peak memory is 12.236 GiB, versus MiniMax-Remover's lowest 10.776 GiB. Appendix G specifies measurement boundaries.

Cost of a second token stream. With the same 32K weights and four transformer forwards, adding a source-token stream increases tokens from 13,346 to 26,594, median time by 2.485×, and incremental peak memory by 1.995× (Table 9). These measurements isolate transformer cost. Additive guidance preserves the video sequence length, while increasing K repeats the predictor. End-to-end timing also includes video-processing overhead.

6 Limitations

DVTE-Bench focuses on synthetic text overlays. Graphic logos, stickers, and preservation of natural scene text require broader paired evaluation. Fully covered backgrounds depend on the video prior when neighboring frames provide no visible evidence. Application examples include residual graphics and local texture artifacts (Appendix E).

7 Conclusion

We introduced DVTE-Bench and DeOverlay for reconstructing scenes beneath video text overlays. DVTE-Bench unifies eight overlay categories in 1,631 paired videos with clean targets, aligned pixel masks, and a common evaluation protocol across overlay layouts, temporal behavior, and appearance.

DeOverlay combines source initialization with visibility-normalized local aggregation and global source guidance. Additive conditioning keeps the source available throughout shared-weight flow updates using one video-token sequence. It outperforms four baselines on all five reconstruction and perceptual metrics. On the 200-video subset, four-step flow exceeds the separately trained direct-residual reference by 1.161 dB in Mask PSNR, while component ablations support both guidance branches. Category-level evaluation further separates the challenges of pixel reconstruction and perceptual quality, with artistic titles and vertical text stressing different metrics. The benchmark and method enable joint study of scene recovery and inference cost.

AI USE STATEMENT

Generative AI tools assisted with language editing, literature search, and comparison. The authors take responsibility for the final data, text, claims, and citations.

ETHICS STATEMENT

DVTE-Bench is constructed from existing video clips by adding synthetic overlays. It may inherit content and demographic biases from the source videos, while automated text planning and OCR screening can introduce errors. Any released materials will retain source attribution and follow the licenses and redistribution terms of the underlying videos. Removing logos or watermarks can erase attribution or alter context; applications should be restricted to authorized editing that respects attribution and privacy.

REPRODUCIBILITY STATEMENT

Section 3 describes paired benchmark construction and regional evaluation, while Section 4 specifies the source-anchored flow representation, training objective, and inference updates. Appendix A details the model, conditioning branches, optimization, and deployment operations. Appendices C and D document metric aggregation and the evaluated benchmark checkpoint, masks, inference steps, and postprocessing; Appendix G gives the runtime protocol. The accompanying figure-provenance records identify the displayed source clips, frames, and crops.

REFERENCES

Junye Chen, Chaowei Fang, Jichang Li, Yicheng Leng, and Guanbin Li. Decouple and couple: Exploiting prior knowledge for visible video watermark removal. *IEEE Transactions on Image Processing*, 34:1192–1203, 2025. doi: 10.1109/TIP.2025.3534033. URL https://doi.org/10.1109/TIP.2025.3534033.

Zheng Chen, Ruofan Yang, Jin Han, Dehua Song, Zichen Zou, Chunming He, Yong Guo, and Yulun Zhang. DiffST: Spatiotemporal-aware diffusion for real-world space-time video super-resolution. *arXiv preprint arXiv:2605.13182*, 2026. URL https://arxiv.org/abs/2605.13182.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2022. doi: 10.1109/TPAMI.2020.3045810. URL https://doi.org/10.1109/TPAMI.2020.3045810.

Zihao Fan, Xin Lu, Jie Xiao, Dong Li, Jie Huang, and Xueyang Fu. IR-Flow: Bridging discriminative and generative image restoration via rectified flow. *Machine Intelligence Research*, 2026. doi: 10.1007/s11633-026-1679-0. URL https://arxiv.org/abs/2604.19680.

Qingdong He, Chaoyi Wang, Peng Tang, Yifan Yang, and Xiaobin Hu. CLEAR: Context-aware learning with end-to-end mask-free inference for adaptive subtitle removal. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65005.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Zheng Hui and Yunlong Bai. SEDiT: Mask-free video subtitle erasure via one-step diffusion transformer. *arXiv preprint arXiv:2605.14894*, 2026. URL https://arxiv.org/abs/2605.14894.

Sangwoo Jo, Donggeun Ko, Jayeon Kang, Youngsang Kwak, Jaehwa Kwak, and Sungjoon Choi. Flow straight to reality: Perceptually consistent flow matching for efficient image restoration. In *Computer Vision – ECCV 2026*, volume 17066 of *Lecture Notes in Computer Science*, pp. 223–239. Springer, 2026. doi: 10.1007/978-3-032-37718-0_14. URL https://link.springer.com/chapter/10.1007/978-3-032-37718-0_14.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2412.08629.

Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. DiffuEraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025. URL https://arxiv.org/abs/2501.10018.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, Yanhong Zeng, Qin Zhao, Zifan Shi, Wei Wu, Hao Ouyang, Qiuyu Wang, Shangzhan Zhang, Jiahao Shao, Yipengjing Sun, Liangxiao Hu, Lunke Pan, Nan Xue, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Ka Leong Cheng. Scaling mixture-of-experts video pretraining for embodied intelligence. *arXiv preprint arXiv:2607.07675*, 2026. URL https://arxiv.org/abs/2607.07675.

Robbyant. LingBot-Video. GitHub repository, 2026. URL https://github.com/Robbyant/lingbot-video. Accessed September 24, 2026.

Mohammadreza Salehi, Mehdi Noroozi, Luca Morreale, Ruchika Chavhan, Malcolm Chadwick, Alberto Gil Couto Pimentel Ramos, and Abhinav Mehrotra. RFDM: Residual flow diffusion models for video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://cvpr.thecvf.com/virtual/2026/poster/38778.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://www.cns.nyu.edu/pub/lcv/wang03-reprint.pdf.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1801.03924.

Shangchen Zhou, Chongyi Li, Kelvin C. K. 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. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhou_ProPainter_Improving_Propagation_and_Transformer_for_Video_Inpainting_ICCV_2023_paper.html.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2536. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6d255ddeb1f4a5ebb8e448b3998bb8e5-Abstract-Conference.html.

Appendix Contents

A Implementation Details . . . 12
B Euler Error and Endpoint Prediction . . . 13
C Metric Definitions and Reporting Protocol . . . 14
D Overlay Categories and Evaluation Coverage . . . 15
E Qualitative Exports and Case Analysis . . . 17
F Ablation Details . . . 17
G Timing and Optimization Protocol . . . 18
H Complete Per-Category Comparison . . . 23
I Temporal Diagnostic Computation . . . 23

A Implementation Details

Training corpus. The training corpus contains 350,000 paired samples: 250,000 base samples, 50,000 stylized-text samples, and 50,000 mirrored-text samples. Each sample provides an overlaid input, its aligned clean target, and an overlay mask. These training samples are separate from all 1,631 DVTE-Bench clips, including its 68 speech-aligned subtitle clips. The three training subsets have sampling weights $4 : 2 : 1$ for base, stylized, and mirrored text, respectively; these weights specify sampling frequencies rather than corpus sizes.

Table 6 gives the multistep configuration.

Table 6: Configured model and optimization settings.

Setting	Value
Backbone	LingBot-Video dense 1.3B configuration
Latent channels / patch size	16 / $(1, 2, 2)$
Transformer blocks / width / heads	24 / 2048 / 16
Feed-forward width / text embedding width	6144 / 2560
LoRA rank / scale	256 / 1.0
Local / global branch hidden width	512 / 512
Local temporal window	5 latent positions
Local mask dilation radius	1 token-grid cell
Loss-mask dilation radius	1 latent-grid cell
Model-mask dropout probability	0.2
Source-endpoint perturbation σ	0.05 (training)
Overlay-region loss coefficient λ	5.0
Pixel-crop / high-frequency loss weights	0 / 0
Optimizer / learning rate	AdamW / 5×10^{-5}
Betas / weight decay	$(0.9, 0.95)$ / 0
Warm-up / maximum configured updates	300 / 60,000
Precision / gradient clipping	bfloat16 / norm 1.0
Preview steps / batch inference default	4 / 6
DVTE-Bench checkpoint / mask / steps	40K / GT / 6
Guidance scale	2.0
VAE posterior mode	Sample

Token-grid conditioning. The transformer averages the source latent over each spatial patch before local context aggregation and max-pools the model mask to the same grid. The temporal patch size is one. The local branch therefore operates on patch-grid source features, while the global branch averages the original latent over all temporal and spatial positions. Mask dilation in the local branch and dilation in the loss use their respective resolutions.

Source-guidance projections. The local projection uses two $1 \times 1 \times 1$ convolutions with hidden width 512:

$$A_{t,p} = \mathrm{dilate}(m)_{t,p}\, W_{A,2}(\mathrm{SiLU}(W_{A,1}\bar{s}_{t,p} + b_{A,1})) + \mathrm{dilate}(m)_{t,p}\, b_{A,2}. \tag{7}$$

Dilation has radius one on the token grid. Padded temporal positions contribute zero to both sums in Eq. 4. The global source branch uses a two-layer MLP with hidden width 512:

$$G(z_s) = W_{G,2}\,\mathrm{SiLU}\left(W_{G,1}\frac{1}{T_\ell H_\ell W_\ell}\sum_{t,p} z_{s,t,p} + b_{G,1}\right) + b_{G,2}. \tag{8}$$

Both guidance output projections and the mask embedding are initialized to zero. Joint video–text self-attention uses three-dimensional rotary embeddings, time-dependent modulation, and dense feed-forward layers. LoRA of rank 256 adapts eligible linear layers; the patch embedding, guidance modules, and mask embedding are trained directly. The VAE and text encoder remain fixed. The VAE uses posterior sampling. Pixel-crop and high-frequency losses have zero weight.

Guidance and deployment. For classifier-free guidance (Ho & Salimans, 2022), positive and negative text conditions give

$$\widehat{v}_\theta = v_\theta^- + g(v_\theta^+ - v_\theta^-). \tag{9}$$

Both evaluations retain the source and conditioning mask. The default $g = 2$ requires two network evaluations per Euler step. The decoder maps the final latent to the output video. Long videos use overlapping chunks, with previous outputs supplying latent boundary conditions. Deployment derives text masks from OCR; removing non-text overlays requires a mask covering the designated graphic region. Full-benchmark evaluation uses 12-pixel seamless pixel paste-back, with the method-specific output processing specified in Appendix D.

Resolution and time sampling. The configured data buckets use 41 frames at a short-side resolution of 768 for videos with original long side at least 1920, 65 frames at 768 for long side at least 1280, and 121 frames at 608 for smaller inputs. The decoder samples a random starting frame with unit frame stride. The training sampler uses the continuous shifted-mode distribution with mode scale 1.1. The configured resolution shift ranges from 1 to 5 over the selected token-count range, and inference also applies the resolution shift to the linear–quadratic schedule.

Training and inference procedure. Algorithm 1 describes one training draw and core inference. For clarity, it omits batch padding and chunk overlap. When overlap conditioning is active, the corresponding latent elements are supplied as conditions and excluded from the target mask q in Eq. 5.

Algorithm 1 DeOverlay training and inference

Require: Paired X, Y, M, instruction $\mathbf{c}$, velocity network v_θ

Training

Encode $z_s \leftarrow \mathcal{E}(X)$ and $z_c \leftarrow \mathcal{E}(Y)$
Sample $\epsilon \sim \mathcal{N}(0, I)$ and continuous time τ
$a \leftarrow z_s + 0.05\epsilon$; $z_\tau \leftarrow (1-\tau)z_c + \tau a$
Set $M_{\text{model}} \leftarrow 0$ with probability 0.2, otherwise M
Predict $v \leftarrow v_\theta(z_\tau, \tau, \mathbf{c}; z_s, M_{\text{model}})$
Update trainable parameters using Eq. 5, target $a - z_c$, and true mask M

Inference

Encode $z_s \leftarrow \mathcal{E}(X)$; set $z^{(0)} \leftarrow z_s$
Construct descending schedule $\tau_0 = 1 > \cdots > \tau_K = 0$
for $k = 0, \ldots, K-1$ **do**
    Evaluate guided velocity $\widehat{v}_\theta$ with fixed source and mask
    $z^{(k+1)} \leftarrow z^{(k)} + (\tau_{k+1} - \tau_k)\widehat{v}_\theta$
end for
return $\mathcal{D}(z^{(K)})$

B Euler Error and Endpoint Prediction

Repeated updates change both the states seen by the predictor and the numerical approximation of its flow. For a known training pair, the displacement in Eq. 2 is constant, and an oracle reaches the clean

latent from the perturbed source endpoint in one step. The network estimates this displacement from the current state and time, allowing its prediction to change along the trajectory.

Consider the learned guided field $f(z, \tau) = \widehat{v}_\theta(z, \tau, \mathbf{c}; z_s, M_{\text{model}})$ with the conditions fixed. Let $z_f(\tau)$ be its exact ODE solution initialized at z_s. If f is Lipschitz in z and sufficiently smooth along this solution, Euler integration on a fixed interval has global error of order h, where $h = \max_k |\Delta\tau_k|$. The endpoint reconstruction error can be decomposed as

$$\|z^{(K)} - z_c\| \le \underbrace{\|z^{(K)} - z_f(0)\|}_{\text{integration error}} + \underbrace{\|z_f(0) - z_c\|}_{\text{learned-flow endpoint error}} . \tag{10}$$

Reducing the maximum step size h tightens the integration-error bound under these assumptions. Endpoint accuracy also depends on the learned field and uncertainty about hidden content, so integration accuracy alone does not establish improved detail. The following derivation gives the error recursion and explains the state-dependent predictor. The shared-checkpoint study in Section 5.3 tests reconstruction quality directly as K varies.

We now derive the standard Euler approximation bound for the learned field. Set $u = 1 - \tau$ and $F(z, u) = -f(z, 1 - u)$, so integration proceeds forward from $u = 0$ to $u = 1$. Assume F is Lipschitz in z with constant L on a region containing the exact and numerical trajectories. Suppose the exact solution satisfies $\|d^2 z_F/du^2\| \le B$ there. Taylor expansion of the exact solution over an interval h_k gives

$$z_F(u_{k+1}) = z_F(u_k) + h_k F(z_F(u_k), u_k) + r_k, \qquad \|r_k\| \le \tfrac{1}{2} B h_k^2. \tag{11}$$

Subtracting this update from Euler and using the Lipschitz condition yields

$$e_{k+1} \le (1 + L h_k) e_k + \tfrac{1}{2} B h_k^2, \qquad e_k = \|z^{(k)} - z_F(u_k)\|. \tag{12}$$

With the same initial condition, $e_0 = 0$. Let $h = \max_k h_k$. Iterating the bound gives, for $L > 0$,

$$e_K \le \frac{Bh}{2L}(e^L - 1), \tag{13}$$

and the $L = 0$ limit is $Bh/2$. Constants L and B are assumptions rather than measured properties of the trained network. This bound concerns approximation of its ODE, not the distance between the ODE endpoint and the clean target.

Why a straight paired path can yield a varying predictor. The population minimizer of an unweighted squared velocity loss is the conditional mean

$$v^*(z, \tau, \xi) = \mathbb{E}[a - z_c \mid z_\tau = z, \tau, \xi], \qquad \xi = (z_s, M_{\text{model}}, \mathbf{c}). \tag{14}$$

For the coordinate-weighted loss in Eq. 5, the corresponding minimizer is the weighted conditional mean, $\mathbb{E}[w_i(a_i - z_{c,i}) \mid z, \tau, \xi]/\mathbb{E}[w_i \mid z, \tau, \xi]$, when the denominator is positive and the coordinate's valid-sample weighting is absorbed into w_i. Although a particular paired displacement is constant, the conditional distribution over displacements can depend on the observed intermediate state. This explains why evaluating the field again can change the predicted correction. It does not ensure that a multimodal hidden background will be recovered faithfully. The matched-checkpoint experiment tests that outcome directly.

C Metric Definitions and Reporting Protocol

Tables 2 and 8 use the complete 1,631-video release and pixel-accurate `mask_packed` GT masks. Scores are transcribed from the released machine-readable records. Except for global VFID and the aggregate TWE gap, frame or frame-pair measurements are averaged within videos and then equally across videos. The finite PSNR mean omits infinite-valued video scores, with their count reported separately.

Pixel and structural fidelity. Mask PSNR is computed from errors over GT mask pixels in each frame. Mask SSIM averages the spatial SSIM map over those pixels, rather than over the enclosing rectangle. Input masks and scoring masks are separate: a pipeline may receive bounding boxes while its output is scored on the common pixel mask.

Perceptual fidelity. For LPIPS and DISTS, form $\widehat{Y}_t^M = M_t \odot \widehat{Y}_t + (1 - M_t) \odot Y_t$, replacing the exterior prediction with clean-target pixels. One temporally stable crop encloses the union of mask locations over the video and is shared by prediction and target. Active-mask frames are resized to a short side of 256 pixels with the long side capped at 512. The same GT-composited crop video supplies RGB-I3D features of dimension 1,024 through the ProPainter evaluator for VFID. VFID is the global Fréchet distance between feature distributions; it is not averaged from per-video or per-type distances. These constructions isolate errors around the overlay and do not measure exterior preservation.

Released temporal diagnostics. Optical flow uses Farneback with maximum image side 320. The temporal region is the union of the current GT mask and the previous mask transported to the current frame by GT backward flow. Flow statistics use the previous mask union the current mask sampled into previous-frame coordinates by GT forward flow. TWE uses the sampling operation defined in Appendix I and restricts error aggregation to the mask region, with RGB errors on the $[0, 255]$ scale. TC, Flow Mean, and Flow Var retain the release's named diagnostics and numerical scale. Table 2 uses this released computation.

The released TWE gap is computed *after* aggregation:

$$\Delta_{\mathrm{TWE}} = \left| \frac{1}{N} \sum_{i=1}^{N} \mathrm{TWE}(\widehat{Y}_i) - \frac{1}{N} \sum_{i=1}^{N} \mathrm{TWE}(Y_i) \right|, \qquad N = 1{,}631. \tag{15}$$

The common target reference is 9.3228. This dataset-level statistic accompanies the target-referenced reconstruction metrics; Appendix I specifies its interpretation.

D Overlay Categories and Evaluation Coverage

Table 7: **DVTE-Bench coverage and complete evaluation.** All released clips are evaluated by five methods. The evaluation protocol is defined in Section 3.3.

Challenge type	Overlay characteristic	Release	Evaluated
Standard subtitle	Lower-third captions and style variation	232	232
Artistic title	Display typography, outlines, and glow	218	218
Moving text	Frame-varying trajectories	232	232
Vertical text	Vertical writing at varied positions	215	215
Non-bottom layout	Placement outside the bottom region	223	223
Multi-region	Distributed simultaneous text regions	225	225
Persistent watermark	Persistent semi-transparent text marks	218	218
ASR subtitle	Speech-aligned captions	68	68
Full release		1,631	1,631

Overlay function and benchmark challenge type describe different properties. Subtitles, text watermarks, titles, and name or role captions describe function; position, motion, opacity, and affected area describe appearance and layout. A vertical name caption, for example, combines a function with a spatial layout. The current benchmark assigns each clip one challenge type. Its clean-target OCR policy excludes confirmed text, so preservation of natural scene text requires a separate evaluation setting.

Dataset composition. The release contains 1,631 clips across eight overlay types; its evaluation protocol is defined in Section 3.3. Figure 6 summarizes the manifest counts. The complete release has 1,085 portrait and 546 landscape clips. Language metadata records Chinese (845), English (350), Korean (201), Japanese (178), Thai (50), Italian (2), and one clip each labeled mixed, Spanish, Portuguese, Polish, and Welsh. These counts describe coverage; the rare labels do not support reliable per-language comparisons. Every target frame is scanned under the OCR rejection policy described in Section 3.

DVTE-Bench organizes overlay conditions by layout, motion, typography, and caption timing. Table 8 reports DeOverlay across all eight types, covering 1,631 clips. Aggregates weight videos equally,

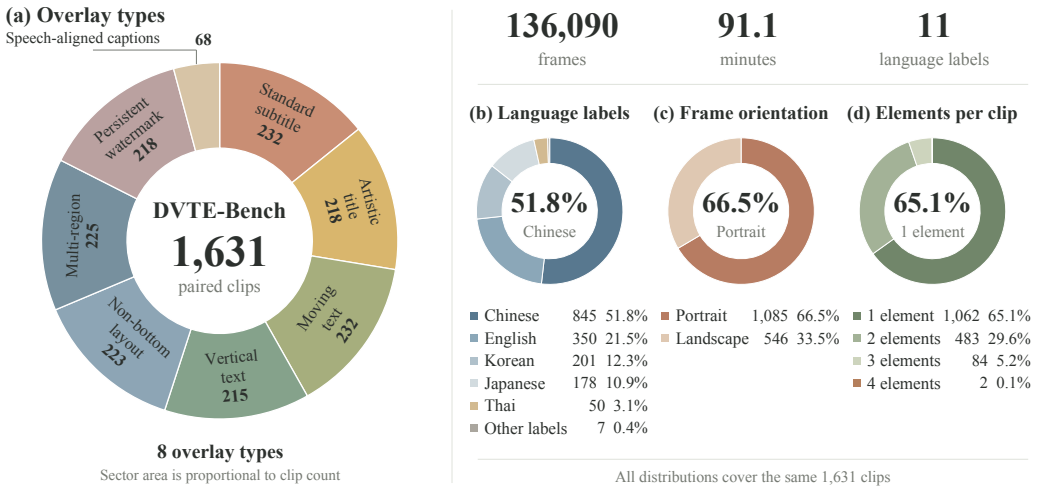

Figure 6: **DVTE-Bench release distributions.** All panels count the 1,631 released clips: (a) assigned overlay type, (b) language metadata, (c) frame orientation, and (d) the per-clip number of rendered overlay elements. "Other labels" combines Italian (2) and mixed, Spanish, Portuguese, Polish, and Welsh (1 each). Overlay-element count does not measure spatial regions or pixel coverage.

rather than assigning equal weight to each type. DeOverlay leads Mask PSNR, SSIM, LPIPS, and DISTS among the five evaluated methods in each type.

Table 8: **DeOverlay across all eight DVTE-Bench types.** All 1,631 clips are included. The 40K checkpoint uses GT bounding-box masks, six steps, and the released seamless paste-back setting. Scoring uses pixel masks; N_∞ counts clips omitted only from finite-mean PSNR.

Text type	Clips	PSNR† ↑	SSIM↑	LPIPS↓	DISTS↓	N_∞
Standard subtitle	232	30.39	0.8718	0.0285	0.0395	1
Artistic title	218	26.24	0.8409	0.0674	0.0668	0
Moving text	232	30.73	0.8800	0.0209	0.0325	4
Vertical text	215	30.46	0.8736	0.0975	0.1015	3
Non-bottom layout	223	30.92	0.8843	0.0414	0.0628	2
Multi-region	225	29.88	0.8834	0.0103	0.0167	2
Persistent watermark	218	36.02	0.9306	0.0314	0.0380	3
ASR subtitle	68	34.20	0.9184	0.0137	0.0527	0

Evaluated pipelines. The benchmark metadata identifies the 40K checkpoint, GT bounding-box masks, six steps, and a 12-pixel seamless paste-back setting for DeOverlay. ProPainter, MiniMax-Remover, and DiffuEraser share GT bounding-box masks and copy source pixels outside an 8-pixel dilated mask. CLEAR uses official mask-free inference. These settings define the evaluated system pipelines; the scoring region remains the dataset GT mask for every method. Appendix C specifies the pixel-mask scoring and perceptual crop protocol. The qualitative exports in Appendix E retain their own run provenance.

Mask provenance is recorded independently of category. An oracle mask measures reconstruction with a known edit region. An estimated mask measures the associated detection pipeline. A manually supplied mask is a separate input setting and will not be labeled automatic detection. For watermarks and captions with graphic elements, coverage must include the entire designated overlay rather than only recognized characters.

E Qualitative Exports and Case Analysis

The application examples in Figure 5 display matched source and DeOverlay output frames. In each pair, the source is on the left and the output is on the right at the same frame index.

Selection and presentation. Figure 5 uses twelve matched source/output crops from the supplied current-model exports. They cover Chinese, English, bilingual, and animation subtitles; name captions and dialogue; vertical and large titles; a text watermark; a translucent caption; a promotional sticker; and a television logo. Figures 8 and 9 retain the corresponding full frames and mark the crop coordinates. Frame indices and coordinates are also recorded in the figure-source manifest and editable figure notes. The television-logo case uses a logo-only mask; the remaining program graphics are outside that requested edit. The large-title source already contains a transition double exposure, visible on both sides. These exports are not a controlled step-count comparison. All operations are frame extraction, equal-coordinate cropping, and display scaling, with no color or contrast enhancement. Across the broader export collection, occasional subtitle remnants, local texture artifacts, and residual geometric watermark elements remain.

Paired benchmark context. Figure 7 extends the five main-paper examples to seven overlay categories, adding a persistent watermark and a standard subtitle. All seven cases have 81 frames. We use zero-based frame 40 and the four-step DeOverlay outputs from the same-A100 timing package. The internal `sedit_4step` filenames refer to DeOverlay, not the unrelated SEDiT paper. CLEAR's native output is fitted without changing its aspect ratio and centered on the source-sized canvas before applying the common crop. The artistic-title crop focuses on the upper title and car; the multi-region crop focuses on the upper overlay, with its location marked in the source thumbnail. The small, translucent watermark is enlarged without contrast adjustment. These selected cases illustrate restoration behavior rather than estimating an aggregate ranking or comparing inference depths.

Multilingual generalization. Figure 10 shows Thai, Indonesian, German, Portuguese, Japanese–Chinese, and Vietnamese–Chinese text overlays. The Thai and Indonesian clips share the same scene content, as do the German and Portuguese clips, making differences in glyph shapes and line layouts visible on matching backgrounds. The bilingual examples combine two writing systems within the same frame. Each row includes two time points with matched source/output coordinates, showing reconstruction beneath changing subtitle content and across different scenes. Background patterns, clothing, printed scene text, and objects remain visible in the displayed outputs.

F Ablation Details

The component and step-count ablations use 200 randomly sampled videos from DVTE-Bench; the main comparison in Table 2 evaluates all 1,631 released videos. The step sweep in Table 4 holds the multistep checkpoint fixed and changes the inference update count. Table 3 uses four steps for the component variants and reports a separately trained, matched one-step direct-residual reference. The quality tables retain the reported Mask MAE, Outside MAE, MaskCrop SSIM, and TWE absolute-gap scales.

F.1 Training Formulation and Initialization

Under the same experimental environment, we independently train a one-step direct-residual baseline to predict the source-to-target residual. In Table 3, four-step flow improves Mask PSNR from 31.300 to 32.461 dB, reduces MAE from 4.520 to 4.299, raises Crop SSIM from 0.91120 to 0.92765, and reduces TWE absolute gap from 0.195 to 0.180. These results compare the two trained models at their respective inference settings. Table 4 studies update count within the continuous-flow checkpoint.

The Gaussian-start model in Table 3 is trained for a noise-initialized, two-video-stream input. Its 2.861 dB gap to the source-initialized model reflects both the starting state and the way source evidence enters the network. The independent token microbenchmark below evaluates the computational effect of the second stream.

F.2 Source Conditioning

The full four-step model has Outside MAE of 2.541. This rises to 2.552 with both source branches removed, 2.543 without VSA, and 2.550 without GSC. These small increases accompany the covered-region changes in Table 3 and describe exterior behavior within the component study. Keeping both branches but replacing visibility-normalized aggregation with an ordinary temporal mean gives Outside MAE of 2.542 and a TWE absolute gap of 0.183, compared with 0.180 for the full model.

F.3 Transformer Token Cost

Table 9: **One versus two video-token streams.** The microbenchmark uses real 32K weights in bfloat16 and a latent tensor of shape $1 \times 16 \times 23 \times 36 \times 64$, corresponding to an 81-frame input with eight mirror-padding frames at the 768s bucket. Both inputs include 98 text tokens. Time covers four transformer forwards and excludes VAE encoding, decoding, and video processing. Memory is incremental peak allocation above the loaded runtime.

Input organization	Video tokens	Total tokens	Time (s)↓	Memory (GiB)↓
Source-initialized stream	13,248	13,346	5.266	1.744
Noise and source streams	26,496	26,594	13.088	3.479

We warm up each input organization and interleave four timed trials, reversing their order between rounds. CUDA synchronization bounds each measurement. The table reports medians; the 10th–90th percentile time ranges are 5.260–5.272 s and 13.077–13.099 s, respectively. Incremental memory subtracts the pre-forward allocation from the peak allocation. The machine can carry concurrent GPU load, so these timings characterize the recorded workload rather than isolated end-to-end deployment speed.

G Timing and Optimization Protocol

Shared input workload. The end-to-end comparison in Table 5 uses eight deterministically selected clips, one per overlay type. Seven clips contain 81 frames at 540×960 resolution; the remaining clip contains 83 frames at 876×492, yielding 650 input frames. Every method processes these same inputs on one otherwise idle A100, in serial order. Each runner retains its internal resizing and processing operations. MiniMax-Remover uses six steps with maximum side 960; DiffuEraser uses maximum side 960, model offloading, and maximum attention slicing.

Timing boundaries and aggregation. One-time model loading is excluded from the per-video measurements. End-to-end time includes each pipeline's decoding, inference, postprocessing, and video encoding. Seconds per frame is the sum of measured seconds divided by 650, and FPS is its reciprocal. The inference-only intervals follow each runner's CUDA synchronization boundaries, so end-to-end time is the primary cross-method comparison. For DeOverlay, the logged inference interval includes the diffusion stage and its associated codec operations, rather than transformer forwards alone.

Memory measurement. A common external sampler queries GPU memory every 200 ms across each method's process. Table 5 reports the largest sampled value in GiB. The one- and four-step DeOverlay runs share one process and therefore share its recorded peak of 12.236 GiB. This measurement includes the loaded runtime and differs from the incremental forward-allocation measurements in Table 9.

Optimization. The configured optimizer is AdamW with learning rate 5×10^{-5}, betas $(0.9, 0.95)$, zero weight decay, 300 warm-up steps, and gradient clipping at 1.0. The configuration uses bfloat16 and a maximum of 60,000 updates.

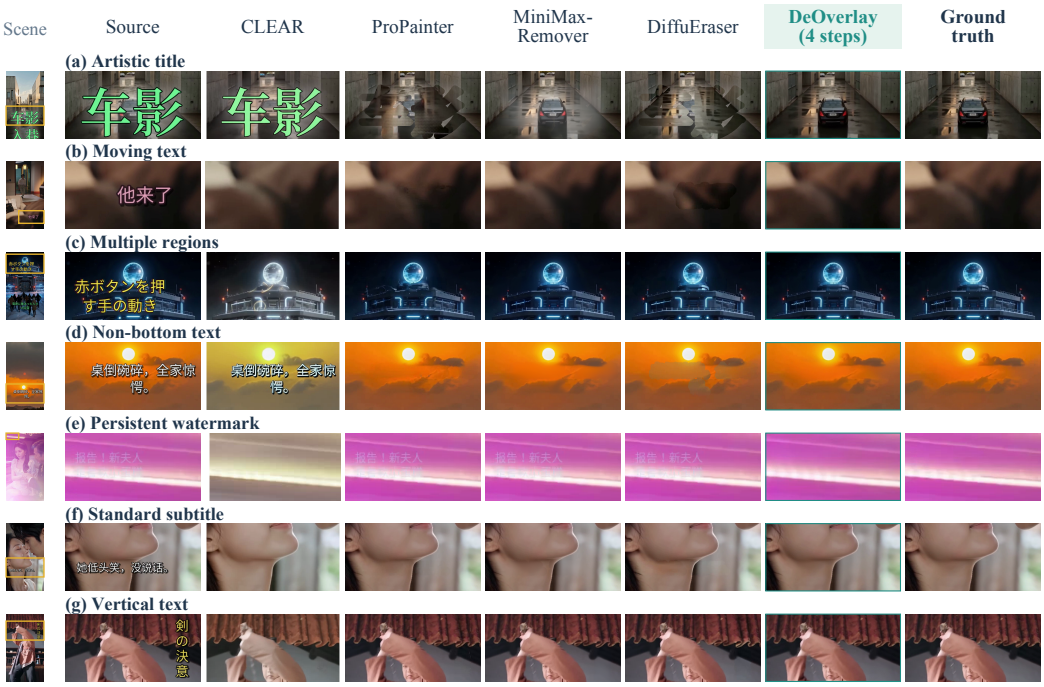

Figure 7: **Expanded comparison across seven overlay categories.** Every row shows an identical frame and common crop for the source, four baselines, DeOverlay ($K = 4$), and clean target. Small source frames locate the crop with a yellow box.

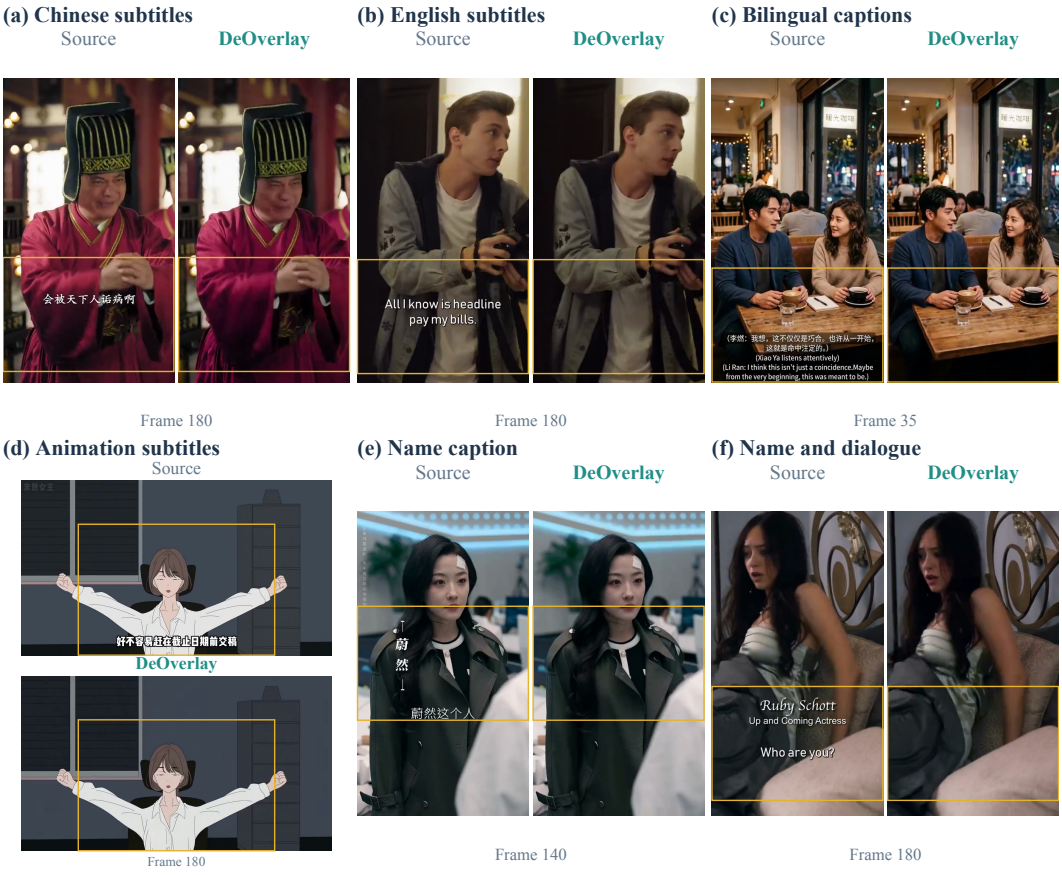

Figure 8: **Application context for Figure 5(a–f).** Adjacent source/output frames preserve native orientation. Yellow boxes locate the matching main-paper crops; frame indices are zero-based.

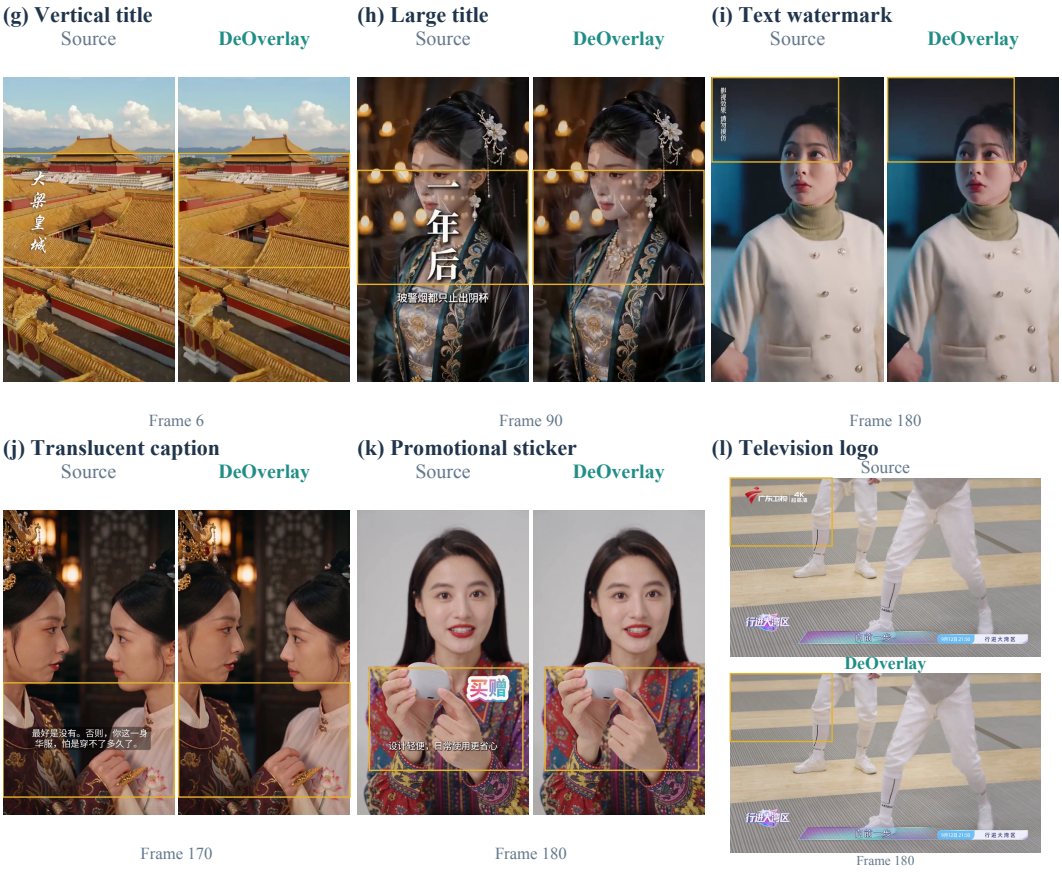

Figure 9: **Application context for Figure 5(g–l).** Full source/output pairs include titles, watermarks, captions, a sticker, and the selected television-logo edit. Unrelated program graphics remain in the logo-only case. Yellow boxes locate the main-paper crops.

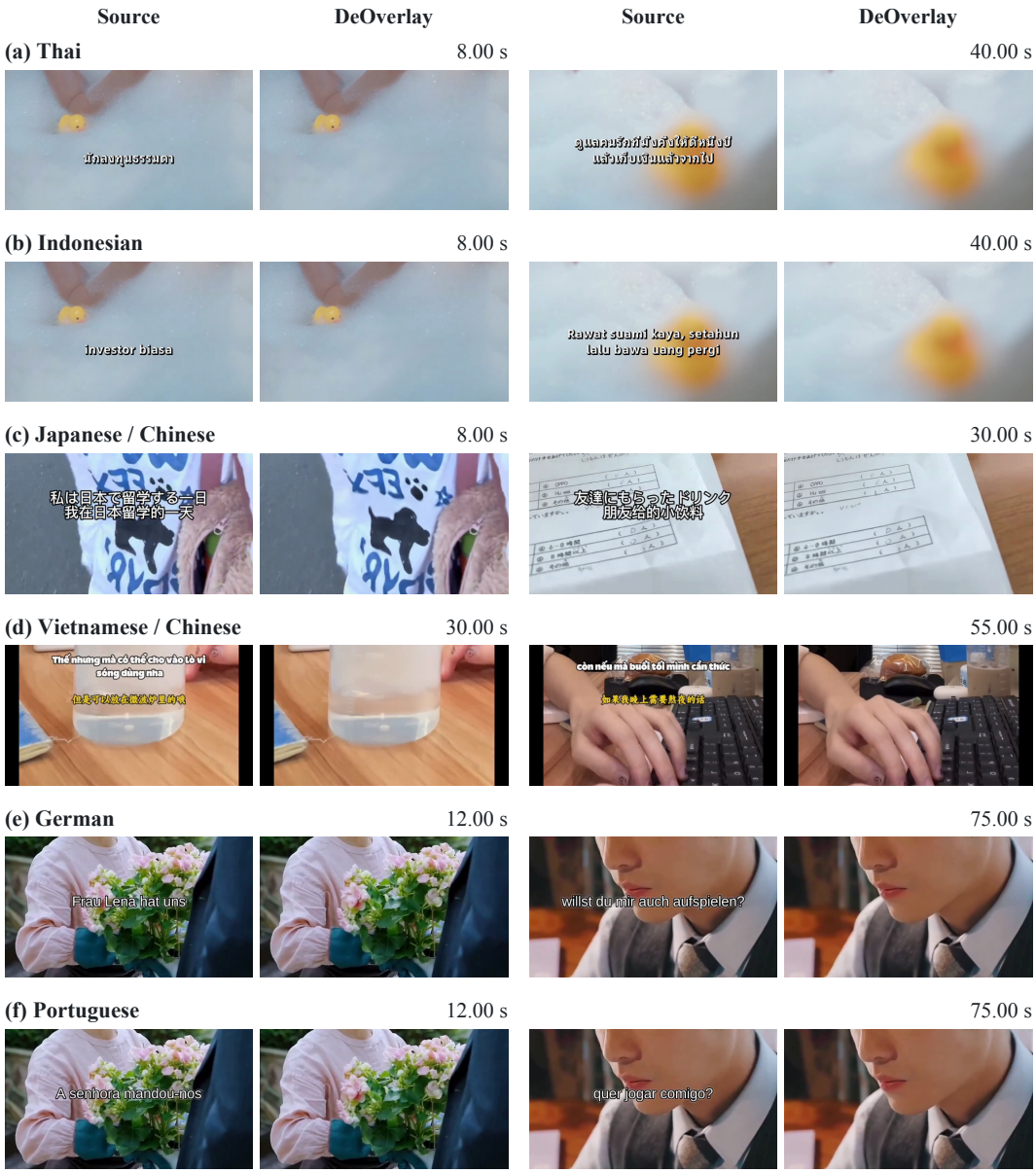

Figure 10: **Multilingual generalization on application videos.** Six language settings include Thai, Indonesian, Japanese–Chinese, Vietnamese–Chinese, German, and Portuguese overlays. Each row contains two matched source/DeOverlay pairs; timestamps identify the displayed frames. Input and output use identical crop coordinates and display scales. These application clips have no clean targets.

H Complete Per-Category Comparison

Tables 10 and 11 give the complete five-method comparison for each overlay type under the protocol in Appendix D.

Table 10: **Reconstruction across overlay categories (1/2).** All five methods are evaluated on every clip in each category, using the same pixel-mask scoring protocol. DeOverlay uses the 40K checkpoint and six steps. PSNR is the finite mean, and N_∞ counts infinite-valued video scores. Bold marks the best quality score within each category.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	N_∞
Standard subtitle (232 clips)					
CLEAR	17.98	0.5726	0.2192	0.1567	0
ProPainter	25.17	0.7668	0.0890	0.0778	1
MiniMax-Remover	24.92	0.7752	0.0838	0.0755	0
DiffuEraser	24.17	0.7501	0.0899	0.0783	1
DeOverlay ($K = 6$)	**30.39**	**0.8718**	**0.0285**	**0.0395**	1
Artistic title (218 clips)					
CLEAR	8.67	0.2878	0.5548	0.3276	0
ProPainter	19.20	0.6639	0.2766	0.1707	0
MiniMax-Remover	22.16	0.7395	0.1677	0.1149	1
DiffuEraser	19.73	0.6903	0.2078	0.1373	2
DeOverlay ($K = 6$)	**26.24**	**0.8409**	**0.0674**	**0.0668**	0
Moving text (232 clips)					
CLEAR	16.28	0.5194	0.1890	0.1451	1
ProPainter	27.61	0.8158	0.0577	0.0569	4
MiniMax-Remover	27.16	0.8194	0.0518	0.0556	0
DiffuEraser	25.97	0.7890	0.0566	0.0592	3
DeOverlay ($K = 6$)	**30.73**	**0.8800**	**0.0209**	**0.0325**	4
Vertical text (215 clips)					
CLEAR	13.10	0.3324	0.5796	0.3836	0
ProPainter	26.92	0.7975	0.1756	0.1432	3
MiniMax-Remover	27.63	0.8275	0.1604	0.1328	1
DiffuEraser	26.55	0.8023	0.1756	0.1430	4
DeOverlay ($K = 6$)	**30.46**	**0.8736**	**0.0975**	**0.1015**	3

I Temporal Diagnostic Computation

Table 2(b) reports the release's temporal measurements on all 1,631 videos. Its TWE label follows the forward-flow remap convention specified in the release metadata. With forward displacement $u_{t-1\to t}$, the referenced computation samples the previous frame as

$$I^{\rightarrow}_{t-1}(x) = I_{t-1}(x + u_{t-1\to t}(x)) \tag{16}$$

and averages its absolute RGB difference from $I_t(x)$ on the scoring region. This is a forward-displacement sampling diagnostic. Backward warping of the previous frame into the current frame instead samples with $u_{t\to t-1}$. The table retains the released diagnostic names and numerical scales. TC, Flow Mean, and Flow Var follow the release's reported labels.

The aggregate gap applies the absolute value after video-level averaging, $|\overline{\mathrm{TWE}}_{\mathrm{pred}} - \overline{\mathrm{TWE}}_{\mathrm{GT}}|$, with clean-target reference 9.3228. Opposite signed deviations can therefore cancel. The target-referenced reconstruction comparison is reported in Table 2.

Table 11: **Reconstruction across overlay categories (2/2).** Settings and notation follow Table 10. Together the two tables cover all 1,631 released clips.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	N_∞
Non-bottom layout (223 clips)					
CLEAR	17.76	0.5941	0.3264	0.2282	0
ProPainter	24.84	0.7627	0.1462	0.1269	2
MiniMax-Remover	25.31	0.7909	0.1195	0.1137	1
DiffuEraser	24.22	0.7582	0.1356	0.1229	2
DeOverlay ($K = 6$)	**30.92**	**0.8843**	**0.0414**	**0.0628**	2
Multi-region (225 clips)					
CLEAR	15.81	0.5273	0.1506	0.1017	0
ProPainter	23.48	0.7634	0.0453	0.0449	2
MiniMax-Remover	24.14	0.7915	0.0361	0.0372	1
DiffuEraser	23.10	0.7613	0.0411	0.0401	2
DeOverlay ($K = 6$)	**29.88**	**0.8834**	**0.0103**	**0.0167**	2
Persistent watermark (218 clips)					
CLEAR	20.96	0.7017	0.2886	0.1835	0
ProPainter	33.09	0.8858	0.0566	0.0543	3
MiniMax-Remover	30.36	0.8570	0.0929	0.0768	0
DiffuEraser	29.23	0.8328	0.1114	0.0888	2
DeOverlay ($K = 6$)	**36.02**	**0.9306**	**0.0314**	**0.0380**	3
ASR subtitle (68 clips)					
CLEAR	20.50	0.7638	0.1992	0.2097	0
ProPainter	29.59	0.8698	0.0443	0.0977	0
MiniMax-Remover	31.28	0.9064	0.0275	0.0764	0
DiffuEraser	28.06	0.8469	0.0520	0.1025	0
DeOverlay ($K = 6$)	**34.20**	**0.9184**	**0.0137**	**0.0527**	0