FORGELOC-R1: SET-LEVEL REINFORCEMENT FINE-TUNING OF VIDEO MLLMS FOR TEMPORAL FORGERY LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal forgery localization predicts every manipulated interval in a partially forged video. Existing methods rely on detectors built for particular manipulation traces, whereas a general-purpose video multimodal large language model (MLLM) could localize any forgery family. Reinforcement fine-tuning with GRPO and an overlap reward is the standard method for temporal grounding, but transplanted to forgery localization it loses its learning signal. We show that both losses are inherent in the group-relative update: a group whose sampled intervals miss or barely overlap a short segment yields rewards that differ only by sampling noise, and the update is biased toward the segment count most frequent in the training data. ForgeLoc-R1 restores the signal at both points while leaving the GRPO update unchanged, and scores each rollout with a set reward that the two mechanisms act on. 1) At the *group level*, anchor-guided advantage pruning inserts the ground-truth answer into every group as an anchor, which we show gives a group without signal a likelihood step toward the ground truth. Reward shaping then bounds the anchor's margin, and only the anchor and the rollouts with the largest advantage magnitudes are trained on. 2) At the *reward level*, a set-level reward counts the distinct segments a rollout has found, so a missed segment and an unmatched prediction both lower it, and ranks the misses by their distance to the ground truth. 3) At the *objective level*, a count-balanced objective weights multi-segment videos so that the policy is not biased toward the most frequent segment count. On ActivityForensics and DDL, ForgeLoc-R1 outperforms the strongest reinforcement fine-tuning baseline by 4.34 and 3.11 mIoU, with the largest gains on multi-segment videos and on a generator held out from training.

1 INTRODUCTION

Recent advances in AI-driven video generation have reshaped content creation, and current models synthesize videos with high realism and fine-grained controllability Kong et al. (2024); Yang et al. (2025). A creator can generate a video from a text description Bao et al. (2024) or from a set of reference frames Blattmann et al. (2023); Zhu et al. (2025), or edit an existing video Jiang et al. (2025); Wu et al. (2023). The barrier to producing realistic video has therefore dropped sharply, and generated video is now common in film, advertising, and social media.

These techniques also create a new problem for content authenticity. A forger can replace a contiguous span of frames in an authentic video with generated content, while the remaining frames stay untouched. The unmodified frames show the original scene, and the replaced span shows an event that did not occur. Such a video keeps the provenance of the authentic video it was made from, and it spreads through the same channels as authentic video. Partial forgeries therefore pose a risk that grows with the quality of the generators Cai et al. (2024); Miao et al. (2025). Temporal forgery localization addresses this risk by predicting start and end times of every replaced span in the video.

Existing methods address temporal forgery localization with dedicated architectures Zhang et al. (2023) or combine detector proposals with MLLM-based refinement Feng et al. (2026b). Video MLLMs such as Qwen2.5-VL Bai et al. (2025) represent temporal predictions as text, and recent work fine-tunes them for temporal grounding with GRPO Shao et al. (2024); Li et al. (2026). GRPO

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.

samples multiple responses per input and normalizes their rewards within the group, so one response is updated according to its reward relative to the others. Forgery localization produces the same output and is scored by the same overlap-based reward, so this paradigm applies to it directly. Its effect, however, depends on the differences among the rewards in a group and on which responses win most groups, and forgery localization breaks both. First, no prompt names the manipulated segment, so the model samples intervals by content. For a short manipulation most sampled intervals miss the ground truth or barely overlap it. The rewards of a group then coincide or differ only by sampling noise, so the normalized advantages carry no information or rank the responses by noise. Second, one segment count dominates the training data, so the answer that reports it earns above-average reward in most groups. The update is then biased toward that count unless the gain on the remaining videos outweighs their rarity.

To address both failures, we propose ForgeLoc-R1, a theoretically grounded GRPO framework for set-valued temporal localization that restores informative group-relative signals and corrects segment-count bias without altering the underlying update rule. The framework operates at three complementary levels: the composition of the group, the reward that scores each rollout, and the weight of each video in the objective. 1) At the *group level*, anchor-guided advantage pruning inserts the ground-truth answer into each group as an anchor. We prove that an anchor with a reward margin induces a directed likelihood update toward the anchor, even when all sampled rollouts miss the target. Reward shaping bounds the anchor's margin so that the sampled rollouts remain ranked, and only the anchor and the rollouts with the largest advantage magnitudes are trained on. 2) At the *reward level*, we introduce a set-level reward whose maximum requires the correct segment count and perfect one-to-one matches. Missed segments and unmatched predictions therefore lower the reward, while non-overlapping predictions remain distinguishable. 3) At the *objective level*, an analysis of segment-count imbalance reveals a bias that drives training toward the most frequent count. This motivates a count-balanced objective that reweights videos to counteract this bias.

Experiments on ActivityForensics and DDL demonstrate the effectiveness of ForgeLoc-R1 for temporal forgery localization. All methods are post-trained from Qwen2.5-VL-7B under matched training conditions. ForgeLoc-R1 outperforms the strongest baseline TempSamp-R1 by 4.34 and 3.11 mIoU on the two benchmarks. The gains are largest on videos with multiple manipulated segments, where recovering all segments is particularly challenging. ForgeLoc-R1 improves by 5.25 mIoU on a generator excluded from training, which indicates better generalization to unseen forgeries.

2 Related Work

Temporal Forgery Localization. Temporal forgery localization predicts the manipulated intervals of a video. Most work solves it with a detector designed for the task. Early detectors localize boundaries with multimodal architectures trained on audio-visual deepfakes Cai et al. (2022); Zhang et al. (2023), and later ones add cross-modal cues such as the mismatch between visual and audio speech Katamneni & Rattani (2024); Koutlis & Papadopoulos (2024). Such a detector is built for the traces of its target manipulations, and a new forgery family requires a new detector. The cost of a new detector has grown as benchmarks moved from manipulated faces to regenerated scenes Miao et al. (2025); Bao et al. (2026). Recent work pairs a dedicated detector with an MLLM, the detector proposing candidate segments and the MLLM refining their boundaries and explaining them Feng et al. (2026b). The MLLM only adjusts what the detector proposes, so the segments that can be found are still set by the detector. We post-train a general-purpose video MLLM to predict the intervals directly, so that one model applies to any forgery family without a detector.

Reinforcement Fine-Tuning. Reinforcement learning has emerged as an effective paradigm for post-training large language models and MLLMs Guo et al. (2025); Team et al. (2025); Deng et al. (2025). Existing approaches are broadly categorized into reward model-based methods and direct preference optimization. Reward model-based methods, such as RLHF Bai et al. (2022) and PPO Schulman et al. (2017), rely on a separately trained reward model to guide policy updates, whereas direct preference optimization Rafailov et al. (2023) bypasses explicit reward modeling and optimizes preference pairs directly. GRPO Shao et al. (2024) extends this paradigm by sampling a group of responses for each input and normalizing their rewards within the group, so that policy updates depend on the relative quality of the responses rather than on absolute reward values. Its optimization signal consequently requires the responses within a group to receive distinguishable

rewards. Multi-segment forgery localization violates this requirement in two ways. Responses sampled for a short segment miss or barely overlap it and receive rewards that differ only by sampling noise, and the update is biased toward the segment count most frequent in the data.

Temporal Video Grounding. Given a language query, temporal video grounding requires a model to localize the corresponding interval in an untrimmed video. SFT has been the predominant approach for adapting MLLMs to this task. TimeChat Ren et al. (2024) binds each frame representation to its timestamp so that the model can report times. However, SFT computes its loss on the text of the annotated answer, whereas localization is evaluated by the overlap between the predicted and the ground-truth interval. RL optimizes this overlap directly, and recent methods apply GRPO with an IoU reward for temporal grounding Li et al. (2025). Video-R1 Feng et al. (2026a) further adds a temporal-consistency bonus through shuffled-frame contrast, while TempSamp-R1 Li et al. (2026) injects ground-truth solutions into the sampling group. In all of these methods the query guides the sampling and designates the single interval against which the reward is computed. Our target is instead a set of segments, and no query indicates where they lie, so Section 3 redefines both the reward and the group.

3 METHOD

Figure 1 illustrates an overview of ForgeLoc-R1, which trains the policy to recover the complete set of manipulated segments in a forged video. We first propose Anchor-Guided Advantage Pruning (§3.2), which inserts a ground-truth anchor into each group and restricts the update to the anchor and the sampled responses with the largest advantage magnitudes. We then define the Set-Level Reward (§3.3) that both mechanisms act on, which counts the distinct segments a rollout has found and ranks the misses by distance. Finally, we introduce the Count-Balanced Objective (§3.4), which weights each training example by its ground-truth segment count, so that the policy is not biased toward the segment count that is most frequent in training data.

3.1 PRELIMINARIES

Group-relative optimization. GRPO Shao et al. (2024) trains a policy by comparing sampled responses within a group. For each input x, the behavior policy $\pi_{\theta_{\text{old}}}$ samples G responses $\{o_j\}_{j=1}^{G}$, and each response receives a scalar reward r_j that measures how well it meets the requirement of the task. The advantage of a response is its reward normalized within the group,

$$A_j = \frac{r_j - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)}. \tag{1}$$

The standard deviation is the sample standard deviation with denominator $G - 1$. The policy is updated by maximizing the clipped surrogate objective

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\tfrac{1}{G}\textstyle\sum_j \min\big(\rho_j A_j,\ \text{clip}(\rho_j, 1 \pm \epsilon) A_j\big)\Big] - \beta_{\text{KL}}\, \mathbb{D}_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big], \tag{2}$$

where $\rho_j = \pi_\theta(o_j \mid x)/\pi_{\theta_{\text{old}}}(o_j \mid x)$ is the importance ratio and π_{ref} is a frozen reference model. Eq. 2 is written at the sequence level. The implementation applies the ratio per token and averages over the tokens of a response, which divides its term by the response length, a factor the analysis omits since the answers have nearly equal length. Two properties follow from the normalization of Eq. 1. First, the update depends only on the differences among the rewards in a group. When these rewards coincide, the input contributes no update, and when they differ only slightly, the small denominator turns slight differences into large advantages. Second, the update sums the advantages over groups, so a response that earns above-average reward in most groups is reinforced unless its losses on the remaining groups outweigh its gains.

Problem Formulation. Given a video in which a forger has regenerated one or more temporal segments, temporal forgery localization requires predicting the manipulated segments as a set $P = \{p_i\}_{i=1}^{K_p}$ of intervals $p_i = (s_i, e_i)$, where K_p is the number of predicted intervals. A prediction is evaluated by matching its intervals to the ground-truth segments $\mathcal{G} = \{g_j\}_{j=1}^{K_g}$. We cast the task as text generation with a general-purpose video MLLM. The model takes as input frames sampled

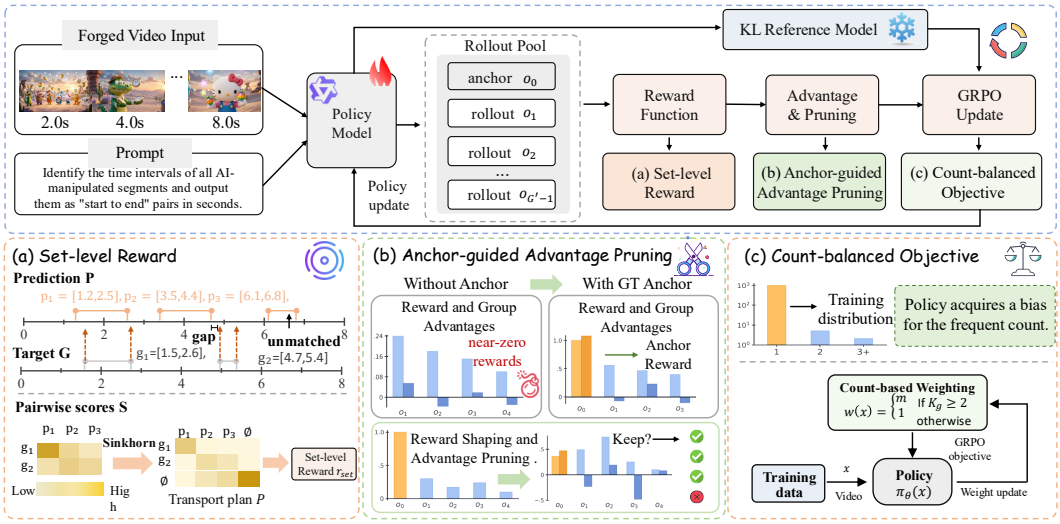


Figure 1: Overview of ForgeLoc-R1. (a) The set-level reward counts the distinct segments found, so a missed segment and an unmatched prediction both lower it, and ranks the misses by distance. (b) Anchor-guided advantage pruning inserts a ground-truth anchor into every group and trains on the anchor and the rollouts with the largest advantage magnitudes. (c) The count-balanced objective gives greater weight to multi-segment videos.

at a fixed rate f and outputs the intervals in seconds as text, *e.g.* `<answer>12.3 to 15.7; 40.2 to 44.5</answer>`. The prompt is given in Appendix A.3. We fine-tune the model with GRPO, taking the video as the input, the answer text as the response, and a reward computed from the overlap between the predicted intervals and the ground-truth segments.

3.2 Anchor-Guided Advantage Pruning

A general-purpose video MLLM does not initially perceive the traces of manipulation, so the intervals it samples are distributed independently of the manipulated segment. Under an IoU reward a miss scores zero and a rollout that barely overlaps the segment scores close to zero, so the rewards of a group sampled for a short segment cluster near zero (Proposition 4). When the rewards of a group coincide, by the first property of Section 3.1 it contributes no gradient. When they differ only by the variation between samplings, the advantages rank the rollouts by noise. Yet Eq. 1 normalizes within the group, so one member whose reward exceeds the rest by a margin restores the learning signal and directs it toward that member. The following proposition gives the gradient of such a group.

Proposition 1 (A directed update from one response with a margin)**.** *Let a group of G' members consist of a response o_0 with reward r_0 and $G'-1$ sampled rollouts with rewards $r_j \le r_0 - \delta$ for some $\delta > 0$. Then the standard deviation in Eq. 1 over the G' members is at least $\delta/\sqrt{G'}$, the advantage A_0 of o_0 is positive, and the gradient of the surrogate objective of Eq. 2 for this group, taken at $\rho_j = 1$ and without the KL term, is proportional to*

$$A_0 \nabla_\theta \log \pi_\theta(o_0 \mid x) + \sum_{j=1}^{G'-1} A_j \nabla_\theta \log \pi_\theta(o_j \mid x). \tag{3}$$

If sampled rewards coincide, then $A_0 = (G'-1)/\sqrt{G'}$ and $A_j = -1/\sqrt{G'}$ for every rollout.

The first term of Eq. 3 is a likelihood step on o_0 with positive weight, and it is present whether the sampled rewards coincide or differ only by noise. The policy cannot yet produce such a response, but the ground truth can. The ground-truth answer has a near-maximal reward whatever the segment length, so in a group sampled for a short segment it exceeds every sampled reward by a wide margin. We therefore serialize the ground-truth set $\mathcal{G}$ into an answer text $\hat{o}(\mathcal{G})$, which we call the anchor, and add it to every group,

$$\mathcal{O} = \{o_0 = \hat{o}(\mathcal{G})\} \cup \{o_j \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)\}_{j=1}^{G'-1}, \tag{4}$$

where G' is the pool size. A group whose rollouts all miss then contributes the likelihood step on the anchor with weight $(G'-1)/\sqrt{G'}$ instead of no gradient.

Reward shaping. Shaping bounds the margin of the anchor so that the differences among the sampled rollouts still show in their advantages. When δ is large, the anchor alone sets the mean and the standard deviation of Eq. 1. Every sampled rollout then receives nearly the same negative advantage, and the differences that rank one rollout above another are suppressed. Once some rollouts overlap the segment, these differences are the signal about its boundaries. We therefore bound δ from above by transforming every reward before normalization,

$$\tilde{r} = \begin{cases} \tau + \beta \log\big(r - \tau + 1\big) & r \geq \tau \\ \tau - \dfrac{e^{\alpha(\tau - r)} - 1}{e^{\alpha} - 1} & r < \tau, \end{cases} \tag{5}$$

where τ is a threshold, β is a small coefficient, and α is a curvature parameter. For $r \geq \tau$ the small β keeps the transformed reward close to τ, which bounds the margin of the anchor over the sampled rollouts. For $r < \tau$ the transform is steepest near $r = 0$, where the sampled rewards concentrate, so their differences are compressed least.

Advantage pruning. Pruning drops the rollouts whose advantages are closest to zero. After the transform, most of the differences among the sampled rollouts are still the variation of the sampled boundaries, which Proposition 5 (Appendix A.1.3) models as zero-mean noise with standard deviation η. Under this model, a rollout whose noise-free reward lies within η of the group mean contributes to the expected gradient at most η/s times its gradient norm while adding the full variance of the noise, as the proposition states. The noise-free reward is not observed, so Eq. 6 selects by the observed $|A_j|$ as a heuristic for this rule and does not inherit the guarantee. We therefore generate the pool of Eq. 4 at size $G' > G$, compute the advantages of all candidates over the full pool after the transform of Eq. 5, and train with these advantages on the anchor and the $G-1$ sampled rollouts with the largest $|A_j|$,

$$\mathcal{K} = \{o_0\} \cup \underset{o_j,\ j \in \{1,\ldots,G'-1\}}{\arg\operatorname{top}^{G-1}} \big|A_j\big|. \tag{6}$$

The advantages are not recomputed after the selection, since the mean and standard deviation of the extremes alone would not describe the group. The anchor is always retained because the transform bounds its advantage and a pure top-$|A_j|$ selection could otherwise exclude it.

3.3 Set-Level Reward

A forged video may contain several manipulated segments, so a rollout predicts a set of intervals while Eq. 1 requires one scalar reward per rollout. The set-level reward supplies this scalar and has to meet two requirements. It must rank misses, and it must count the distinct segments a rollout has found. Scoring each prediction by the segment it overlaps most fails the count, since two predictions of one segment both score high. Scoring each segment by the prediction that overlaps it most fails it too, since one long interval covering two segments scores on both. We therefore match predictions and segments one to one Carion et al. (2020), as the evaluation does, and normalize the matched score by the sizes of both sets,

$$r_{\text{set}} = \frac{2 \sum_{(i,j) \in \pi} S_{ij}}{K_p + K_g}, \qquad \pi = \underset{\pi' \in \Pi(K_p, K_g)}{\arg\max} \sum_{(i,j) \in \pi'} S_{ij}, \tag{7}$$

where $\Pi(K_p, K_g)$ is the set of one-to-one assignments and $S_{ij} \in [0,1]$ is the score of the pair (i,j). The matched score counts the distinct found segments, and the normalization makes r_{set} the F1 of the assignment. The following proposition states how r_{set} responds to a missed segment and to a spurious prediction.

Proposition 2 (r_{set} peaks at the true count)**.** *Let a video contain $K_g \geq 2$ ground-truth segments. r_{set} equals one only when $K_p = K_g$ and every matched pair scores one. A prediction or a segment left unmatched lowers r_{set} through the denominator, and adding a prediction raises r_{set} if and only if it raises the matched sum of the optimal assignment by more than $r_{set}/2$. In particular, a rollout with fewer predictions gains from adding one that pairs with a missed segment at a score above $r_{set}/2$, a threshold below $1/2$ as shown in Appendix A.1.4.*

Whether an attempt clears this threshold before it overlaps the segment depends on the pairwise score S_{ij}, which also has to rank misses, the first requirement. IoU fails both. Every prediction that misses a segment scores zero, so a group of misses cannot be ranked. A prediction placed next to a segment without touching it scores zero as well, so the attempt gains nothing. We therefore score each pair by the one-dimensional generalized IoU Rezatofighi et al. (2019) rescaled to $[0, 1]$. This score is positive for two disjoint intervals and decreases as they move apart, so misses are ranked by their distance to the segment. A prediction adjacent to a missed segment scores $1/2$ and thus clears the threshold of Proposition 2.

Because S_{ij} depends on the ratio of overlap to union, the same boundary error in seconds lowers it less on a longer segment than on a shorter one. We therefore add two boundary terms on the pairs of a one-to-one assignment Carion et al. (2020). The term r_{bnd} decays with the absolute start and end errors on a time scale σ, and r_{tiou} is the timestamp-aware IoU of Time-R1, which discounts the IoU of a pair by its start and end errors relative to the video duration. Both are aggregated as the F1 of their assignment, like r_{set}. An excess prediction does not always lower r_{set}, since it can raise the matched sum, so we additionally discount r_{set} when the number of predicted intervals exceeds K_g. The total reward is

$$r = r_{\text{set}} \cdot e^{-\lambda\,(K_p - K_g)_+} + r_{\text{bnd}} + r_{\text{tiou}}, \qquad r_{\text{bnd}} = \frac{2}{K_p + K_g} \sum_{(i,j)\in\hat{\pi}} e^{-\left(|s_i - s_j^g| + |e_i - e_j^g|\right)/\sigma}, \quad (8)$$

where $\hat{\pi}$ is a one-to-one assignment computed by the Hungarian algorithm, $(K_p - K_g)_+ = \max(K_p - K_g, 0)$ counts the excess predictions, λ sets the discount, and σ is a time scale.

3.4 Count-Balanced Objective

The number of manipulated segments differs from video to video. On a video with the most frequent count, a rollout that reports this count tends to outscore one that reports another, since a missed segment and an unmatched extra prediction both lower r_{set}. By the second property of Section 3.1 the update therefore favors the answer with the most frequent count. The following proposition, proved in Appendix A.1.5, gives the condition for the expected reward with two counts.

Proposition 3 (Count bias of the expected reward and its removal by the weight)**.** *Let a fraction $\rho \in (0, 1)$ of the training videos contain two segments and the rest one, and let a policy attempt a second segment on every video with the same probability $q \in [0, 1]$. Suppose the attempt lowers the expected reward of Eq. 8 by $c > 0$ on a single-segment video and raises it by $g > 0$ on a two-segment video, with c and g independent of q. Then the expected reward increases in q if and only if $\rho/(1-\rho) > c/g$, and if $\rho/(1-\rho) < c/g$ it is maximized at $q = 0$, where the policy predicts one segment on every video. For the expected reward with the two-segment videos weighted by $m > 0$, the condition becomes $m\rho/(1-\rho) > c/g$.*

The proposition concerns the expected reward and motivates the reweighting. It does not characterize the gradient of the group-normalized objective of Eq. 2, whose scale within a group is set by Eq. 1. By Proposition 3 the expected reward favors the more frequent count once the other count is too rare for its gain to outweigh the cost that attempting it incurs on the majority. The bias arises from how often each count is sampled and not from the reward within a group, where all rollouts share the same video. We therefore weight the rare counts more heavily in the objective and leave the reward unchanged. When one segment is the most frequent count, the weight applies to every video with at least two segments,

$$\mathcal{J}_w(\theta) = \mathbb{E}_x\big[\, w(x)\, \mathcal{J}(\theta; x)\,\big], \qquad w(x) = \begin{cases} m & K_g(x) \geq 2 \\ 1 & \text{otherwise,} \end{cases} \quad (9)$$

implemented as repeat-factor sampling (Gupta et al., 2019) so that the per-step group size is unchanged. In the model, the weighted expected reward favors the second attempt once m exceeds $(1-\rho)c/(\rho g)$. We keep $m\rho$ below $1-\rho$, so that multi-segment videos keep a smaller weighted share than single-segment videos and the attempt is favored only when its gain exceeds its cost, and choose m from the share of multi-segment videos. Since c and g are not known in advance, this does not ensure $m\rho g > (1-\rho)c$, and the choice is verified by ablation (Section 4.2).

4 EXPERIMENTS

Implementation details. We post-train Qwen2.5-VL-7B (Bai et al., 2025) from the released instruction-tuned checkpoint with full-parameter updates and no LoRA or external data. Training runs under ZeRO-3 (Rajbhandari et al., 2020) on 8×H200 GPUs for 4 epochs with learning rate 10^{-6} and completions of at most 512 tokens. Videos are pre-encoded under a budget of 3584 visual tokens of 28^2 pixels, with frames sampled at 2 fps on AF and at the native 30 fps on DDL. On AF the model reports intervals in seconds. On DDL, where segments last about half a second, all compared methods read frame indices overlaid on the frames (Wu et al., 2025) and report intervals as indices, converted to seconds before rewards and metrics are computed. Each group draws $G'=8$ candidates at temperature 1.0 and trains on $G=4$ of them. The assignment of Eq. 7 uses the relaxation of Appendix A.1.6 with $\varepsilon=0.1$, 60 iterations and a slack score of 0.05, the discount $\lambda=0.5$, and the boundary terms of Appendix A.1.7 $\sigma=0.3$ s on both corpora. The count weight is $m=3$ on AF and $m=2$ on DDL, set under the bound of Section 3.4 from multi-segment shares of 18% and 27%. At inference the model decodes greedily.

Benchmarks and metrics. We evaluate on ActivityForensics Bao et al. (2026) and DDL Miao et al. (2025). AF regenerates segments of several seconds inside authentic videos with six generation and editing models. We train on 3,824 videos, test on 1,565, and hold out one generator (*vidu*) from training. DDL contains word-level lip-sync and face manipulations lasting about half a second, with 6,001 training and 1,500 test videos. We report mIoU, the best IoU of each ground-truth segment with any prediction averaged over segments and videos, and F1 at IoU thresholds $\{0.3, 0.5, 0.7\}$ under a one-to-one matching (Appendix A.2). Average precision is not reported, since it requires a confidence-ranked list of proposals and each method outputs one unordered set of intervals without confidence scores.

Baselines. We compare with the zero-shot base model and three reinforcement fine-tuning methods, GRPO Shao et al. (2024), Video-R1 Feng et al. (2026a), and TempSamp-R1 Li et al. (2026). Each method is trained from the same base model under the same setting, with the reward and advantage design of its released implementation. The methods are adapted to set outputs with an overlap reward normalized over both the predicted intervals and the ground-truth segments, so missing segments are penalized as well.

4.1 MAIN RESULTS

Table 1 reports the main results. ForgeLoc-R1 achieves the best score on every metric of both benchmarks. On ActivityForensics it obtains 64.27% mIoU and 53.05% F1@0.7, outperforming the strongest baseline TempSamp-R1 by 4.34% and 3.22%, and on DDL 54.03% mIoU and 44.47% F1@0.7, outperforming it by 3.11% and 3.87%. The gains over the two baselines without an anchor are far larger, 9.98% mIoU over Video-R1 and 12.99% over GRPO on ActivityForensics and 20.78% and 22.14% on DDL, where both stay below 34% mIoU and their F1@0.7 falls below the zero-shot base model. Without an anchor, a group sampled for a video with a sub-second segment rarely contains a rollout that overlaps it well, so its rewards differ only by sampling noise (Section 3.2). On DDL the gains over TempSamp-R1 in F1@0.3, F1@0.5 and F1@0.7 are 6.32%, 5.21% and 3.87%, all larger than the 3.11% gain in mIoU, so the improvement comes from more segments being localized above the thresholds rather than from tighter boundaries on segments already localized.

4.2 ABLATION STUDY

We analyze our method on ActivityForensics to assess the impact of the key components in ForgeLoc-R1, including the anchor-guided advantage pruning, the set-level reward, and the count-balanced objective. Results demonstrate that these mechanisms jointly enable the policy to report more of the manipulated segments and to place their boundaries more precisely.

Impact of the anchor-guided advantage pruning. To evaluate the anchor-guided advantage pruning, we remove it and train with standard GRPO groups, in which every rollout is sampled from the policy and all rollouts are back-propagated. As indicated in Table 2, mIoU falls from 64.27% to 58.61%, and the drop in F1 grows with the threshold, from 3.71% at 0.3 to 5.61% at

Table 1: **Main results.** All methods share the backbone, data, readout and training budget. The last row gives the gain of ForgeLoc-R1 over TempSamp-R1, the strongest baseline.

	ActivityForensics				DDL			
Method	mIoU↑	F1@.3↑	F1@.5↑	F1@.7↑	mIoU↑	F1@.3↑	F1@.5↑	F1@.7↑
Qwen2.5-VL-7B	22.36	32.44	19.02	9.84	28.38	22.55	21.96	21.53
GRPO	51.28	72.20	60.39	40.26	31.89	41.08	24.35	14.25
Video-R1 Feng et al. (2026a)	54.29	74.72	62.26	41.70	33.25	46.32	25.70	13.65
TempSamp-R1 Li et al. (2026)	59.93	77.36	68.77	49.83	50.92	58.70	51.45	40.60
ForgeLoc-R1 (ours)	**64.27**	**81.87**	**71.98**	**53.05**	**54.03**	**65.02**	**56.66**	**44.47**
Δ vs. TempSamp-R1	(+4.3)	(+4.5)	(+3.2)	(+3.2)	(+3.11)	(+6.3)	(+5.2)	(+3.9)

Table 2: **Component ablation** on ActivityForensics.

Method	mIoU↑	F1@.3↑	F1@.5↑	F1@.7↑
ForgeLoc-R1 (ours)	**64.27**	**81.87**	**71.98**	**53.05**
− Anchor-guided pruning	58.61 (−5.66)	78.16 (−3.71)	67.65 (−4.33)	47.44 (−5.61)
− Set-level reward	57.25 (−7.02)	79.61 (−2.26)	66.91 (−5.07)	48.05 (−5.00)
− Count balance	59.53 (−4.74)	78.77 (−3.10)	68.60 (−3.38)	47.30 (−5.75)

0.7. Section 3.2 accounts for both drops. Without the anchor, the policy learns only the segments that its own rollouts happen to overlap, and a segment they never overlap is never learned, which lowers mIoU on every video. Without the pruning, the rollouts whose advantages are determined by sampling noise enter the update and displace predicted boundaries in random directions.

Impact of the set-level reward. To evaluate the set-level reward, we replace r_{set} of Eq. 7 with the per-prediction best-overlap average, which also removes the count weight. As presented in Table 2, mIoU falls from 64.27% to 57.25%, the largest drop in the table. Under this reward an omitted segment costs nothing and a duplicate of a found segment still scores, so the policy reports only the segments it localizes accurately (Section 3.3).

Impact of the count-balanced objective. To evaluate the count-balanced objective, we train with w=1 while keeping the set reward. As illustrated in Table 2, mIoU falls from 64.27% to 59.53% and F1@0.7 from 53.05% to 47.30%. This drop corresponds to the analysis in Section 3.4. Without the weight, the single-interval answer earns above-average reward in most groups and is reinforced at every step. The policy therefore learns to report fewer segments than the video contains.

4.3 Analysis

Analysis of segment-count behaviour. Table 3 examines whether each policy reports the correct number of segments. On ActivityForensics ForgeLoc-R1 has the lowest under-reporting rate, 40.5% against 54.0% for TempSamp-R1 and 95.9% for GRPO, and the highest complete rate, 44.7%

Table 3: **Segment-count behaviour** (%). Match is the fraction of test videos whose number of predicted intervals equals the number of ground-truth segments. Under is the fraction of multi-segment videos with fewer predictions than segments. Over is the fraction of single-segment videos with more than one prediction. Complete is the fraction of multi-segment videos in which every ground-truth segment is matched at IoU>0.5.

	ActivityForensics				DDL			
Method	Match↑	Under↓	Over↓	Complete↑	Match↑	Under↓	Over↓	Complete↑
GRPO	**80.6**	95.9	**1.9**	3.4	68.7	86.8	8.6	0.0
Video-R1	79.5	79.7	6.6	14.1	71.2	90.4	5.1	0.0
TempSamp-R1	78.7	54.0	10.9	34.4	74.6	**58.3**	8.2	7.8
ForgeLoc-R1 (ours)	79.8	**40.5**	9.5	**44.7**	**78.4**	65.0	**3.9**	**13.2**

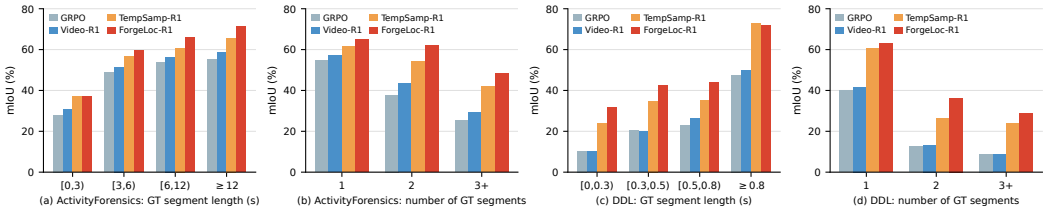


Figure 2: mIoU by ground-truth segment length and by number of ground-truth segments on ActivityForensics and DDL.

Table 4: **Strict-threshold F1** on both benchmarks.

	ActivityForensics		DDL	
Method	F1@.85↑	F1@.95↑	F1@.85↑	F1@.95↑
GRPO	21.02	5.03	10.40	6.73
Video-R1	21.97	5.88	9.16	5.67
TempSamp-R1	27.05	9.12	30.00	23.37
ForgeLoc-R1 (ours)	**30.73**	**10.41**	**31.51**	**23.53**

against 34.4% and 3.4%. Its over-prediction rate of 9.5% stays below the 10.9% of TempSamp-R1, so it reports more of the manipulated segments without predicting more spurious ones. On DDL it has the highest match rate, 78.4% against 74.6% for TempSamp-R1, the lowest over-prediction rate, 3.9% against 8.2%, and the highest complete rate, 13.2% against 7.8% and none for GRPO and Video-R1. These gains stem from the set-level reward with its discount on excess predictions and from the count-balanced objective, which counters the bias toward the most frequent count.

Analysis of segment count and length. Figure 2 groups the test videos by the number and the length of their ground-truth segments. On ActivityForensics the margin of ForgeLoc-R1 over TempSamp-R1 is 7.8% mIoU on multi-segment videos against 3.6% on single-segment videos, and it grows with segment length from a tie below 3 s to 5.8% above 12 s. On DDL the margin is 9.9% on multi-segment videos against 2.4% on single-segment videos, and it is concentrated on segments shorter than 0.8 s, where ForgeLoc-R1 leads by 7.4% to 8.7%. The larger margin on multi-segment videos arises from the set-level reward and the count-balanced objective, which lead the policy to report more of the manipulated segments. The concentration of the margin on short segments arises from the more precise boundaries of ForgeLoc-R1.

Analysis of strict thresholds. Table 4 raises the IoU threshold to 0.85 and 0.95, where a boundary error of a few frames decides the score. At F1@0.85 ForgeLoc-R1 leads TempSamp-R1 by 3.68% on ActivityForensics and 1.51% on DDL. At F1@0.95 the margins shrink to 1.29% on ActivityForensics and 0.16% on DDL, where segments last a few frames and a single-frame error already exceeds the tolerance. The boundary term of Section 3.3 and the pruning of Section 3.2 therefore improve boundary placement up to the resolution of the frame sampling and not beyond it.

5 Conclusion

We present ForgeLoc-R1, a reinforcement fine-tuning method that post-trains a general-purpose video MLLM to localize every manipulated segment of a forged video without a dedicated detector. On this task the standard GRPO method loses its learning signal at two points, since the rollouts sampled for a short segment miss it and earn rewards that differ only by sampling noise, and since the most frequent segment count biases the update. ForgeLoc-R1 restores the signal while leaving the GRPO update unchanged. Anchor-guided advantage pruning inserts the ground-truth answer into every group and trains only on the anchor and the rollouts with the largest advantage magnitudes, a set-level reward scores a rollout by the ground-truth segments it matches one to one and ranks the misses by distance, and a count-balanced objective raises the weight of multi-segment videos. Experiments on ActivityForensics and DDL confirm the effectiveness.

AI USE STATEMENT

We used Claude and Chatgpt for writing assistance, including grammar correction, wording refinement, and minor stylistic editing of draft text. The model was not used for research ideation, data collection, dataset annotation, code generation, experimental design, or analysis. All technical content was authored and verified by the human authors, who take full responsibility for the content of this paper.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

Peijun Bao, Anwei Luo, Gang Pan, Alex C Kot, and Xudong Jiang. Activityforensics: A comprehensive benchmark for localizing manipulated activity in videos. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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.

Zhixi Cai, Kalin Stefanov, Abhinav Dhall, and Munawar Hayat. Do you really mean that? content driven audio-visual deepfake dataset and multimodal method for temporal forgery localization. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–10. IEEE, 2022.

Zhixi Cai, Shreya Ghosh, Aman Pankaj Adatia, Munawar Hayat, Abhinav Dhall, Tom Gedeon, and Kalin Stefanov. Av-deepfake1m: A large-scale llm-driven audio-visual deepfake dataset. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 7414–7423, 2024.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Marco Cuturi. Lightspeed computation of optimal transportation distances. *Advances in Neural Information Processing Systems*, 26(2):2292–2300, 2013.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*, 2025.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026a.

Yue Feng, Jingjing Li, Qijia Lu, Wei Ji, Jingrou Zhang, Fei Shen, Xiao Li, Yizhen Jia, Qiang Chen, Limin Wang, et al. Explainable forensics of manipulated segments in untrimmed long videos. *International Conference on Machine Learning (ICML)*, 2026b.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351–5359. IEEE, 2019.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202. IEEE, 2025.

Vinaya Sree Katamneni and Ajita Rattani. Contextual cross-modal attention for audio-visual deepfake detection and localization. In *2024 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–11. IEEE, 2024.

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.

Christos Koutlis and Symeon Papadopoulos. Dimodif: Discourse modality-information differentiation for audio-visual deepfake detection and localization. *arXiv preprint arXiv:2411.10193*, 2024.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

Yunheng Li, Jing Cheng, Shaoyong Jia, Hangyi Kuang, Shaohui Jiao, Qibin Hou, and Ming-Ming Cheng. Tempsamp-r1: Effective temporal sampling with reinforcement fine-tuning for video llms. *Advances in Neural Information Processing Systems*, 38:40692–40716, 2026.

Changtao Miao, Yi Zhang, Weize Gao, Zhiya Tan, Weiwei Feng, Man Luo, Jianshu Li, Ajian Liu, Yunfeng Diao, Qi Chu, et al. Ddl: A large-scale datasets for deepfake detection and localization in diversified real-world scenarios. *arXiv preprint arXiv:2506.23292*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323. IEEE, 2024.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7589–7599. IEEE, 2023.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13754–13765. IEEE, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Rui Zhang, Hongxia Wang, Mingshan Du, Hanqing Liu, Yang Zhou, and Qiang Zeng. Ummaformer: A universal multimodal-adaptive transformer framework for temporal forgery localization. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 8749–8759, 2023.

Tianyi Zhu, Dongwei Ren, Qilong Wang, Xiaohe Wu, and Wangmeng Zuo. Generative inbetweening through frame-wise conditions-driven video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27968–27978. IEEE, 2025.

A Appendix

A.1 Analysis of the Training Signal

This section analyzes in a stylized model why the standard GRPO method provides little training signal on short manipulated segments and biases the policy toward the most frequent segment count, and how the three components of ForgeLoc-R1 change the signal. The model isolates the structure of the reward and the objective, not the training dynamics of the network.

Setting. A video is normalized to the unit interval. A rollout is a set of K_p intervals $p_i = (s_i, e_i)$, and the ground truth is a set of K_g intervals g_j. For an interval p and a segment g, $\mathrm{IoU}(p, g)$ denotes their IoU, $S_{ij} \in [0, 1]$ denotes the rescaled generalized IoU of Section 3.3 for the pair (i, j), and r_{set} denotes the set-level reward of Eq. 7. Advantages follow Eq. 1 and are set to zero when all rewards in a group coincide.

A.1.1 Short Segments and the Vanishing Group Advantage

Assumption 1. *A video contains one manipulated segment of length ℓ whose centre lies at least $(\ell + L)/2$ from either end of the video. A rollout predicts one interval of length $L \geq \ell$, which may extend beyond the ends of the video. The centre of the interval is uniformly distributed over $[0, 1]$ and independent of the segment location, because the sampled boundaries are determined by the content of the video and the manipulated segment is semantically continuous with its context.*

Proposition 4 (Zero-gradient groups on short segments)**.** *Under Assumption 1 and an IoU reward, the predicted interval overlaps the segment with probability $\ell + L$ and its IoU with the segment is at most ℓ/L. In a group of G independent rollouts, every reward is therefore zero with probability $(1 - \ell - L)^G$, and every advantage in the group is then zero. In every group, all rewards lie in $[0, \ell/L]$.*

Proof. Two intervals of lengths ℓ and L overlap if and only if their centres are closer than $(\ell+L)/2$. For a centre uniformly distributed over $[0, 1]$ this event has probability $\ell + L$, since the segment centre is at least $(\ell + L)/2$ from either end. The intersection of two intervals is at most the shorter length and their union at least the longer, so the IoU is at most ℓ/L. Independence across rollouts gives $(1 - \ell - L)^G$ for the probability that every rollout misses, and Eq. 1 assigns zero advantage to a group whose rewards coincide.

As the segment shortens, the probability $(1-\ell-L)^G$ of a group without gradient increases toward $(1-L)^G$, and the reward range of the groups with an overlapping rollout shrinks in proportion to ℓ. Since Eq. 1 divides by the within-group standard deviation, a small reward range does not make the advantages small. It makes the ranking of the rollouts depend on differences of a few frames of overlap, which are dominated by the randomness of the sampled boundaries discussed in Section 3.2.

A.1.2 The Anchor and the Restored Group Advantage

Proof of Proposition 1. The variance of the G' rewards is a convex function of the $G'-1$ sampled rewards and is invariant under their permutation. The constraint set $r_j \le r_0 - \delta$ is convex and has the same invariance, so the minimum over it is attained with all sampled rewards equal to some $t \le r_0 - \delta$. With this configuration the group mean is $(r_0 + (G'-1)t)/G'$ and the variance equals $(r_0 - t)^2/G'$, which increases with $r_0 - t$, so the minimum over the constraint set is at $t = r_0 - \delta$ and equals δ^2/G'. The reward of o_0 exceeds the group mean, so $A_0 > 0$. At $\rho_j = 1$ the clip in Eq. 2 is inactive, so the summand of rollout j in Eq. 2 equals $\rho_j A_j$ near θ_{old} and its gradient is $A_j \nabla_\theta \rho_j = A_j \nabla_\theta \log \pi_\theta(o_j \mid x)$, which gives Eq. 3 up to the factor $1/G'$. When the sampled rewards coincide at t, the deviations from the mean are $(G'-1)(r_0-t)/G'$ for o_0 and $-(r_0-t)/G'$ for each sampled rollout, and dividing by the standard deviation $(r_0-t)/\sqrt{G'}$ gives $A_0 = (G'-1)/\sqrt{G'}$ and $A_j = -1/\sqrt{G'}$.

A.1.3 Advantage Pruning and the Variance of the Update

Proposition 5 (Bounded bias and reduced variance of advantage-based selection)**.** *Let the reward of rollout j be $r_j = \mu_j + \xi_j$, where μ_j is the part set by its localization quality and ξ_j is zero-mean noise of variance η^2 from the randomness of the sampled boundaries, independent across rollouts and of the log-likelihood gradient $\nabla_\theta \log \pi_\theta(o_j \mid x)$, which is treated as fixed. Suppose the group mean and standard deviation in Eq. 1 are fixed at $\bar{\mu}$, the mean of the μ_j, and at a constant s. Dropping a rollout with $|\mu_j - \bar{\mu}| \le \eta$ from the update changes the expected gradient by at most $(\eta/s)\,\|\nabla_\theta \log \pi_\theta(o_j \mid x)\|$ in norm and removes a variance contribution of $(\eta^2/s^2)\,\|\nabla_\theta \log \pi_\theta(o_j \mid x)\|^2$.*

Proof. The contribution of rollout j to the group gradient is $A_j \nabla_\theta \log \pi_\theta(o_j \mid x)$ with $A_j = (\mu_j - \bar{\mu} + \xi_j)/s$. Since the noise has zero mean, the expectation of this contribution is $(\mu_j - \bar{\mu})/s$ times the log-likelihood gradient of o_j, and under the stated condition its norm is at most η/s times the norm of that gradient. Since the noise has variance η^2 and is independent across rollouts, the variance of the contribution is η^2/s^2 times the squared norm of that gradient, and dropping the rollout removes exactly this term from the variance of the group gradient.

Eq. 6 does not observe μ_j and selects by the observed $|A_j|$, which depends on the noise, so the proposition does not apply to the selected rollouts directly. It motivates the selection by showing that a rollout near the group mean adds variance to the update at little expected gain. The selection is a heuristic based on this analysis, and Table 7 evaluates it.

A.1.4 Multiple Segments and the Shape of the Set Reward

Proof of Proposition 2. An unmatched interval or segment adds to the denominator and not to the matched sum, so it lowers r_{set}. Since every score is at most one and the assignment has at most $\min(K_p, K_g)$ pairs, $r_{\text{set}} \le 2\min(K_p, K_g)/(K_p + K_g)$, which is below one for $K_p \ne K_g$, so $r_{\text{set}} = 1$ requires $K_p = K_g$ with every matched score equal to one. Let M be the matched sum of the optimal assignment. With one more prediction the optimal sum is $M + \Delta M$ for some $\Delta M \ge 0$, and $2(M+\Delta M)/(K_p + K_g + 1)$ exceeds $2M/(K_p + K_g)$ if and only if $\Delta M(K_p + K_g) > M$, that is $\Delta M > r_{\text{set}}/2$. Pairing the new prediction with a missed segment at score b extends the previous assignment, so $\Delta M \ge b$ and $b > r_{\text{set}}/2$ suffices. Since $r_{\text{set}} < 1$ whenever a segment is missed, this threshold is below $1/2$, and for a rollout with a single prediction it is at most $1/(1+K_g) \le 1/3$.

A.1.5 The Most Frequent Segment Count and the Objective

Proof of Proposition 3. The expected reward is $(1-\rho)(r_1 - cq) + \rho(r_2 + gq)$, where r_1 and r_2 are the expected rewards at $q = 0$ on single-segment and two-segment videos and do not depend on q. Its derivative in q is $\rho g - (1-\rho)c$, which is positive if and only if $\rho/(1-\rho) > c/g$. When the derivative is negative the expected reward decreases in q and is maximized at $q = 0$. Weighting the two-segment term by m replaces ρ by $m\rho$ in the derivative.

A.1.6 Entropic Relaxation of the Assignment

The reward used in training replaces the exact assignment of Eq. 7 by an entropic optimal-transport relaxation (Cuturi, 2013). The matrix $\bar{S}$ is S with a slack row and a slack column appended, both filled with a constant κ. The real rows and columns carry unit mass, the slack row carries $\max(1, K_g - K_p)$ and the slack column $\max(1, K_p - K_g)$, and we write $\bar{a}$ and $\bar{b}$ for these row and column masses. Their totals differ by one when $K_p \neq K_g$. Starting from $\exp(\bar{S}/\varepsilon)$, N rounds of normalization in the log domain, each consisting of a row step followed by a column step, give the plan $P_{\varepsilon,N}$, and the training reward is

$$r_{\text{set}} = \min\Big(1,\ \frac{2\sum_{i \le K_p,\, j \le K_g} (P_{\varepsilon,N})_{ij} S_{ij}}{K_p + K_g}\Big), \qquad N = 60. \tag{10}$$

The plan is used as is and is not rounded to an assignment. Scaling all row masses by a common factor changes only the row step and is undone by the column step, so for fixed ε the iterates converge to the entropic optimizer P^*_ε with the effective marginals $(\gamma\bar{a}, \bar{b})$, where $\gamma = \sum_j \bar{b}_j / \sum_i \bar{a}_i$. Eq. 10 therefore differs from Eq. 7 in the entropy term, in the slack, in the rescaled masses and in the finite number of rounds, and Proposition 2 refers to the exact assignment of Eq. 7. As ε approaches zero, P^*_ε approaches the optimal plan of the transport problem with the same marginals and no entropy term when that plan is unique. This limit concerns the converged optimizer rather than $P_{\varepsilon,N}$, and it does not in general recover the assignment of Eq. 7, since the rescaled masses let one interval be assigned to several segments. Table 8 demonstrates that lowering ε from 0.1 to 0.01 changes mIoU by less than one point.

A.1.7 Boundary Terms

Both boundary terms of Eq. 8 are computed on the pairs of a one-to-one assignment found by the Hungarian algorithm and aggregated as $2\sum/(K_p + K_g)$ over the matched pairs. For r_{bnd} the assignment maximizes the matched IoU and the pair score is $e^{-(|s_i - s_j^g| + |e_i - e_j^g|)/\sigma}$. For r_{tiou} the pair score is $\text{IoU}(p_i, g_j)\,(1 - |s_i - s_j^g|/T)_+\,(1 - |e_i - e_j^g|/T)_+$ with T the video duration, and the assignment maximizes this score. A rollout without any prediction scores zero on both terms.

A.2 Evaluation Metrics

Each method outputs one unordered set of intervals per video without confidence scores, so we use the metrics of temporal grounding rather than those of detection. The IoU of two intervals is the length of their intersection divided by the length of their union.

mIoU. For every ground-truth segment we take its largest IoU with any predicted interval of the video, which is zero when the video has no prediction, and average these values over the segments of the video. mIoU is the mean of this per-video value over the test set. It measures how well the ground-truth segments are covered and does not penalize spurious predictions. Figure 2 and Table 6 report the same per-video value grouped by segment length, segment count and generator.

F1 at an IoU threshold. Predicted intervals and ground-truth segments are matched one to one by the assignment that maximizes their total IoU. A matched pair is a true positive if its IoU exceeds the threshold τ. Precision divides the number of true positives by K_p, recall divides it by K_g, and F1@τ is their harmonic mean averaged over videos. A video without any prediction scores zero. Table 1 uses $\tau \in \{0.3, 0.5, 0.7\}$ and Table 4 uses $\tau \in \{0.85, 0.95\}$. Unlike mIoU, F1 penalizes both missed segments and spurious predictions.

Segment-count behaviour and boundary error. Tables 3 and 5 use the same one-to-one assignment. Match, Under and Over compare the number of predicted intervals with the number of ground-truth segments, and Complete counts the multi-segment videos in which every ground-truth segment is matched at IoU above 0.5. The boundary errors of Table 5 are computed on the matched pairs with positive IoU, and the last column is the share of these pairs whose start and end errors are both below the tolerance.

A.3 Prompt

The same prompt is used for training, for all compared method and at inference. The two corpora differ only in the output specification, which asks for seconds on ActivityForensics and for frame indices on DDL.

ForgeLoc-R1: Training and Inference Prompt

System. You are a video forensics expert specializing in detecting AI-generated content.

User. `{sampled video frames}`
The given video contains one or more segments that have been manipulated by AI video generation models — either by INSERTING synthetic activities or by REMOVING/REPLACING original activities through AI inpainting. Other parts of the video are authentic.
Your task: identify the time intervals of ALL AI-manipulated segments.
Look for cues such as:

- Temporal inconsistency: identity drift, attribute flicker across frames
- Unrealistic motion or physics, frozen/looping motion
- Boundary artifacts: warping, blur, or abrupt transitions at the start/end of a segment
- Texture / lighting inconsistencies, unnatural shadows
- Object disappearance or sudden replacement (for delete-type manipulation)

Output specification, ActivityForensics. Output ALL manipulated intervals inside `<answer> </answer>` using the format `"start1 to end1; start2 to end2"` (semicolon-separated, in seconds, precise to two decimal places). If only one segment is manipulated, output a single interval.
Examples: One manipulated segment: `<answer> 2.80 to 14.30 </answer>`. Two manipulated segments: `<answer> 10.50 to 16.40; 24.90 to 42.40 </answer>`

Output specification, DDL. The frames of this video are numbered starting at 0; each frame shows its index at the top-left corner. Output ALL manipulated intervals inside `<answer> </answer>` as FRAME INDICES using the format `"start1 to end1; start2 to end2"` (semicolon-separated, integers). If only one segment is manipulated, output a single interval.
Examples: One manipulated segment: `<answer> 35 to 47 </answer>`. Two manipulated segments: `<answer> 12 to 20; 88 to 101 </answer>`

A.4 Additional Results

Boundary error. Table 5 separates whether a ground-truth segment is matched from how precisely its boundaries are placed. On ActivityForensics ForgeLoc-R1 matches 87.2% of the segments against 81.7% for TempSamp-R1, and its median start and end errors of 0.80 s and 1.04 s are the smallest of all methods, about two sampled frames at 2 fps. On DDL it matches 73.5% of the segments against 76.1% for TempSamp-R1, but its median errors of 0.053 s and 0.120 s are again the smallest, and 38.8% of its matched pairs place both boundaries below 0.1 s against 35.0% for TempSamp-R1 and below 5% for GRPO and Video-R1.

Results per generator. Table 6 breaks the results down by generator on ActivityForensics and by manipulation type on DDL. ForgeLoc-R1 obtains the highest mIoU on five of the six generators and on both manipulation types, and trails TempSamp-R1 by 0.39% only on *ltx*. Its largest margins over TempSamp-R1 are 7.32% on *wan*, where every baseline scores lowest, and 6.75% on *scifi*. On *vidu*, which is excluded from training, the margin is 5.25% and exceeds the 4.34% margin on the full test set, so the gain does not depend on traces specific to the training generators.

Candidate pool size. Table 7 varies the pool size G' while keeping $G = 4$ rollouts in the update. Setting $G' = G$ removes the selection of Section 3.2 and costs 2.22% mIoU, since the rollouts near the group mean then enter the update. Enlarging the pool to $G' = 16$ changes mIoU by 0.74%,

Table 5: **Boundary error on matched pairs.** Matched is the share of ground-truth segments matched one to one with positive IoU. $|\Delta s|$ and $|\Delta e|$ are median absolute boundary errors in seconds, and the last column is the share of pairs with both errors below the tolerance.

	ActivityForensics				DDL			
Method	Matched↑	$\lvert\Delta s\rvert$ ↓	$\lvert\Delta e\rvert$ ↓	<0.5 s↑	Matched↑	$\lvert\Delta s\rvert$ ↓	$\lvert\Delta e\rvert$ ↓	<0.1 s↑
GRPO	72.9	1.15	1.56	6.4	65.3	0.360	0.580	4.6
Video-R1	78.0	1.20	1.47	7.7	67.1	0.400	0.500	3.2
TempSamp-R1	81.7	0.83	1.10	10.8	**76.1**	0.067	0.160	35.0
ForgeLoc-R1 (ours)	**87.2**	**0.80**	**1.04**	**11.7**	73.5	**0.053**	**0.120**	**38.8**

Table 6: **mIoU per generator** on ActivityForensics and per manipulation type on DDL. *vidu* is held out from training.

	ActivityForensics						DDL	
Method	fcvg	ltx	scifi	vace-1.3B	wan	vidu	favf	ravf
GRPO	57.49	63.12	49.95	46.68	41.67	48.84	33.71	30.42
Video-R1	59.66	65.77	53.02	49.09	46.04	52.19	35.35	31.56
TempSamp-R1	68.36	**69.57**	62.33	52.08	48.51	58.76	51.03	50.84
ForgeLoc-R1 (ours)	**72.38**	69.18	**69.08**	**55.12**	**55.83**	**64.01**	**54.10**	**52.18**

within the seed band, so a pool of twice the trained group already supplies enough rollouts whose advantages exceed the sampling noise.

Entropy coefficient. Table 8 lowers the entropy coefficient of the matching from 0.1 to 0.01. mIoU changes by 0.78 points, so the set reward is not sensitive to the strength of the entropy term.

Table 7: **Candidate pool size** on ActivityForensics with $G{=}4$ rollouts back-propagated.

Pool size	mIoU↑	F1@.3↑	F1@.5↑	F1@.7↑
$G'{=}G{=}4$	62.05 (−2.22)	78.61 (−3.26)	69.42 (−2.56)	49.39 (−3.66)
$G'{=}8$ **(ours)**	**64.27**	**81.87**	**71.98**	**53.05**
$G'{=}16$	63.53 (−0.74)	81.00 (−0.87)	71.78 (−0.20)	51.30 (−1.75)

Table 8: **Entropy coefficient** of the matching on ActivityForensics.

Entropy coefficient	mIoU↑	F1@.3↑	F1@.5↑	F1@.7↑
$\varepsilon{=}0.1$ **(ours)**	**64.27**	**81.87**	**71.98**	**53.05**
$\varepsilon{=}0.01$	63.49 (−0.78)	80.26 (−1.61)	71.61 (−0.37)	51.88 (−1.17)

Qualitative examples. Figures 3 to 5 show predictions on ActivityForensics and DDL. In these examples the baselines merge adjacent segments, drop the second segment of a video, or place a single interval away from the forgery, whereas ForgeLoc-R1 reports the number of segments and their extents.



Figure 3: **Qualitative comparison** on ActivityForensics (left) and DDL (right). Red borders mark frames inside a forged segment. The baselines merge, drop or misplace forged segments, whereas ForgeLoc-R1 recovers their number and extent.

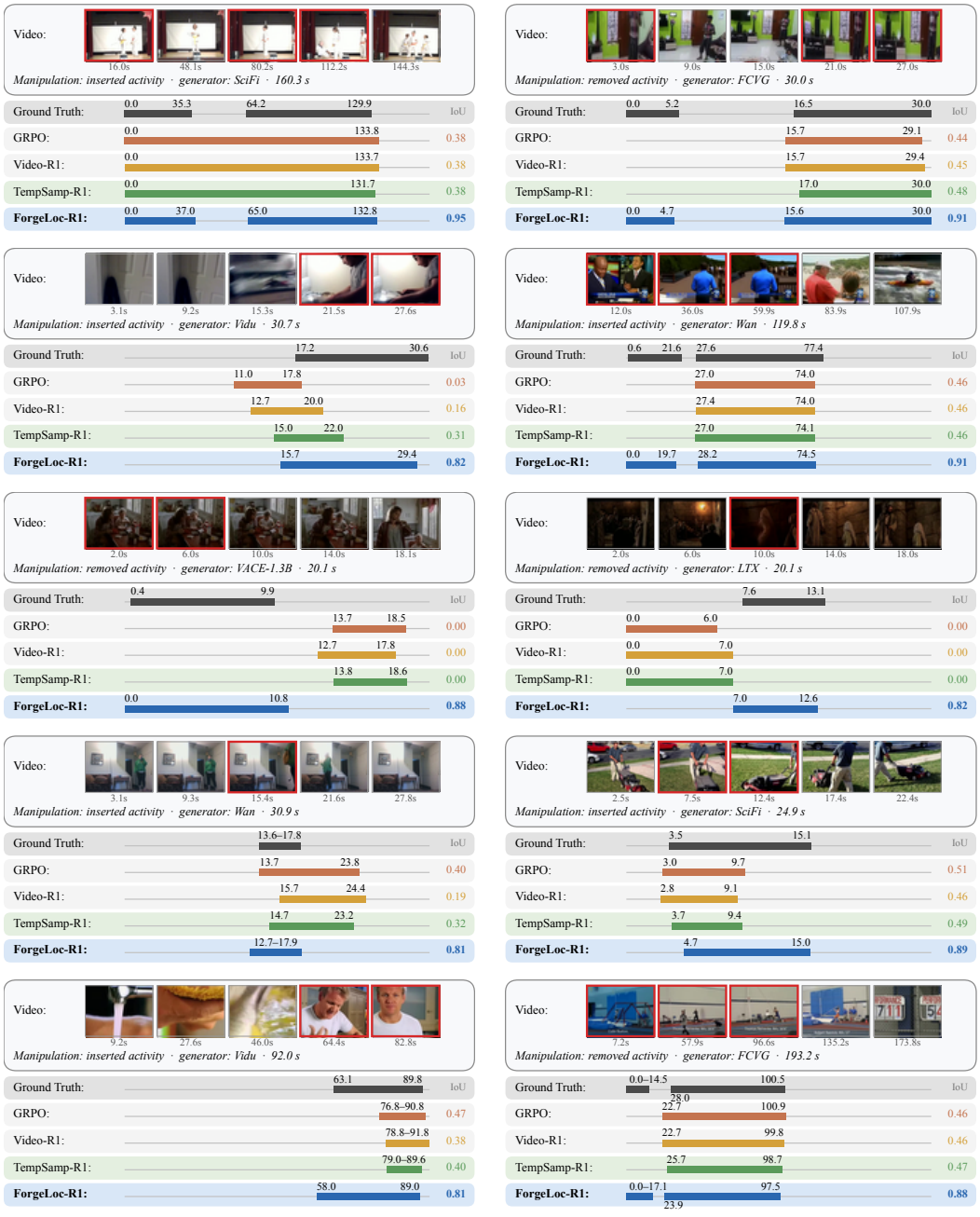


Figure 4: **Qualitative examples on ActivityForensics.** Red borders mark frames inside a forged segment. Bars show ground-truth and predicted intervals, and the number at the right is the video's IoU.

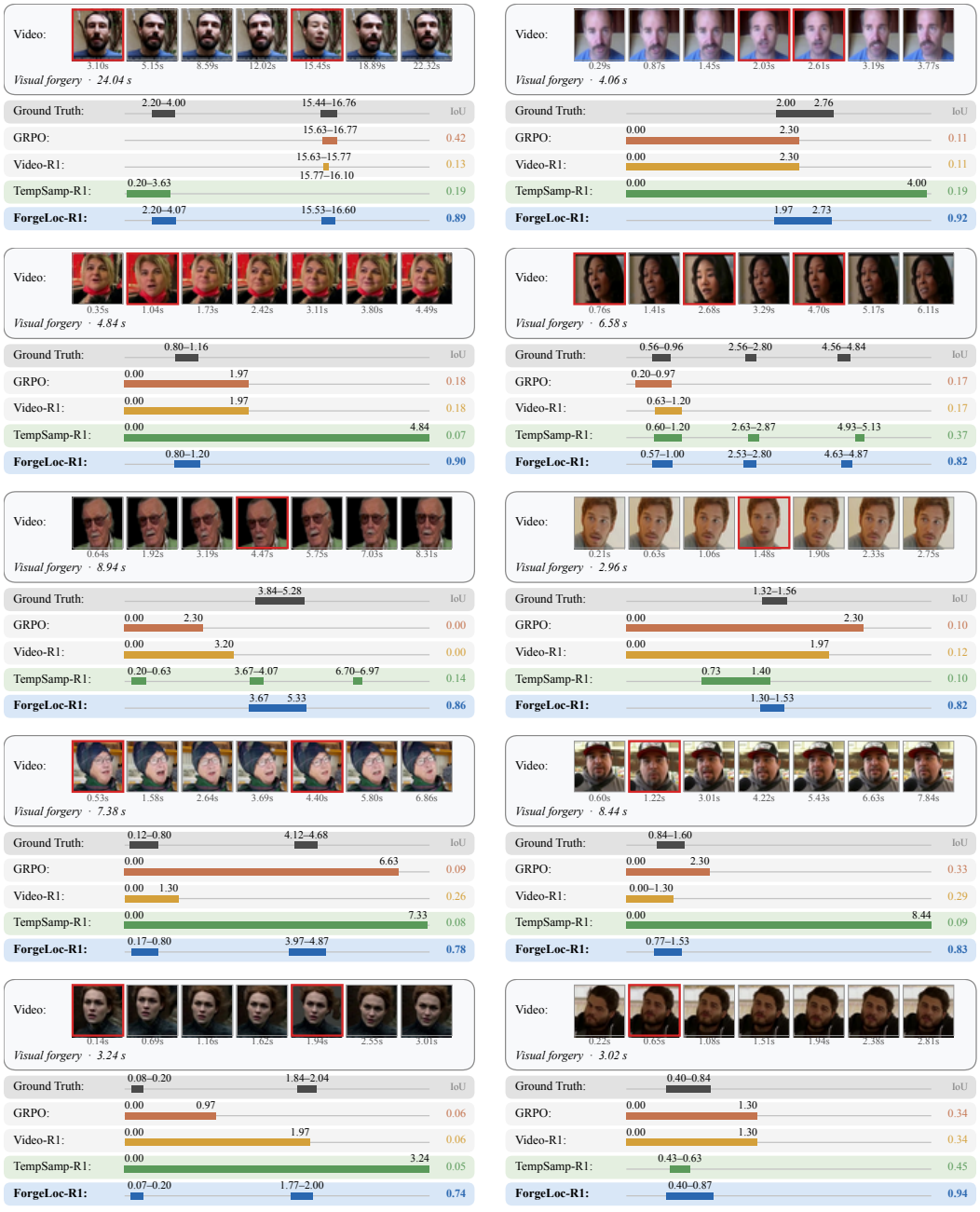


Figure 5: **Qualitative examples on DDL.** Red borders mark frames inside a forged segment. Bars show ground-truth and predicted intervals, and the number at the right is the video's IoU.