

MLT-Dedup: Efficient Large-Scale Online Video Deduplication via Multi-Level Representations and Spatial-Temporal Matching

David Yuchen Wang*
david.w@tiktok.com
TikTok, Singapore

Haoying Li*
haoying.li@tiktok.com
TikTok, Singapore

Hailun Xu
xuhailun@tiktok.com
TikTok, Singapore

Wei Chee Yew
weichee.yew@tiktok.com
TikTok, Singapore

Zirui Zhu
zirui.zhu@tiktok.com
TikTok, Singapore

Sanjay Saha
sanjay.saha@tiktok.com
TikTok, Singapore

Hao Hei
hao.hei1@tiktok.com
TikTok, Singapore

Kanchan Sarkar
kanchan.sarkar@tiktok.com
TikTok, U.S.

Kun Xu
daniel.chen28@tiktok.com
TikTok, U.S.

Abstract

The explosive growth of user-generated video content on online platforms is accompanied by the emergence of numerous near-duplicate videos—videos that are identical or highly similar but differ by partial edits. These duplicates degrade user experience and increase storage and bandwidth costs, making large-scale video deduplication a critical task. Existing video deduplication frameworks face a fundamental challenge in retrieving sufficient high-quality candidates under a limited index budget, as well as trade-offs between efficiency and precision. To address these issues, we propose **MLT-Dedup**, an efficient large-scale online video deduplication framework with **Multi-Level** representations and **spatial-Temporal** matching. Our approach employs a **Multi-Level Video Encoder (ML-VE)** to extract both fine-grained frame-level and sparse clip-level embeddings: sparse embeddings support efficient candidate retrieval, while fine-grained embeddings are loaded for precise pairwise matching. During matching, we introduce **DiF-SiM**, a **Differential Feature-enhanced Similarity Module** capable of locating duplicated temporal segments and providing reliable similarity evidence to support policy-driven deduplication decisions. Extensive experiments on a real-world large-scale platform demonstrate that MLT-Dedup reduces online repetition rates by **91%** under 90% precision. Furthermore, our sparse retrieval design achieves a **5×** increase in index size, enabling broader candidate coverage in real-world deployment.

CCS Concepts

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Artificial intelligence**; **Machine learning**.

*Equal contribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD, Jeju, KR

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2026/08
<https://doi.org/XXXXXXXX.XXXXXXX>

Keywords

Video deduplication, video representation, large-scale video retrieval, video copy localization

ACM Reference Format:

David Yuchen Wang, Haoying Li, Hailun Xu, Wei Chee Yew, Zirui Zhu, Sanjay Saha, Hao Hei, Kanchan Sarkar, and Kun Xu. 2026. MLT-Dedup: Efficient Large-Scale Online Video Deduplication via Multi-Level Representations and Spatial-Temporal Matching. In *Proceedings of (KDD)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

1 Introduction

With the proliferation of multi-source user-generated content, a large number of near-duplicate videos emerge, differing only by lightweight transformations such as clipping, watermark insertion, or format conversion [27, 36]. These duplicated videos degrade search and recommendation quality while increasing copyright risks and storage overhead, making video deduplication critical at scale. In practice, video deduplication frameworks consist of three key components: video representation, candidate retrieval, and video pair matching [3, 23, 39]. Specifically, videos are encoded into visual representations, indexed for scalable retrieval, and further examined by matching models to make deduplication decisions. In large-scale user-generated-content (UGC) platforms, deduplication systems must operate under strict index resources while meeting strong requirements on duplicate reduction, as inefficient designs directly reduce candidate coverage and deduplication accuracy. In this paper, we propose an efficient large-scale video online deduplication framework, MLT-Dedup, to address the index size issue while reducing video repetition via multi-granularity representation and spatial-temporal matching. In the video representation stage of deduplication frameworks, visual embeddings are typically generated by a recall model and can take two forms: frame-level and clip-level embeddings [3, 20, 23, 25, 39]. Frame-level embeddings extract features for every frame, providing fine-grained visual details that benefit accurate duplicate detection. Yet under a fixed indexing budget, such dense representations severely limit index capacity, resulting in fewer retrievable candidates and degraded recall. Clip-level embeddings summarize video segments through key-frame selection, or direct clip encoding. While clip-level embeddings significantly reduce memory footprint and indexing cost,

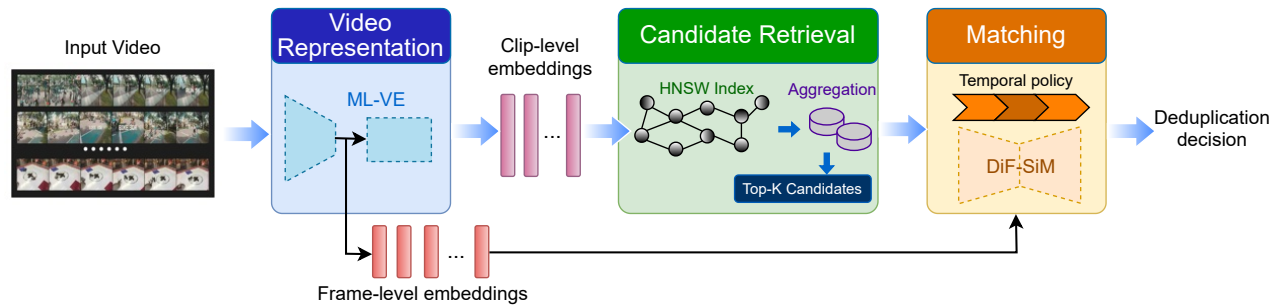


Figure 1: Overall framework of MLT-Dedup: Input videos are encoded by ML-VE into fine-grained frame-level embeddings (used for precise pairwise matching) and sparse clip-level embeddings (indexed via HNSW for high-recall candidate retrieval). Retrieved top-K candidates are matched using DiF-SiM, which computes temporal overlaps to make final deduplication decisions.

their coarse granularity may discard fine temporal details, potentially degrading matching accuracy. This trade-off reveals a key challenge in large-scale deduplication frameworks. To address this challenge, we deploy a multi-level video encoder, ML-VE, which can generate frame-level and clip-level embeddings simultaneously. Coarse clip-level embeddings are used for scalable indexing and retrieval, while fine-grained frame-level embeddings are stored on disk and only loaded for matching when a candidate is retrieved. To ensure effective retrieval performance, ML-VE is trained with dedicated optimization strategies, including hybrid loss formulations, inner-feature knowledge distillation and data-centric memory bank deduplication.

In the video pair matching stage, video duplicate detection methods relying on global similarity [1, 22, 42] degrade as video editing techniques become more sophisticated. They cannot provide temporal localization of duplicated content and are prone to misclassifying partially overlapping videos due to lack of fine-grained temporal information and sensitivity to irrelevant content. To tackle this issue, we introduce the proportion of overlapping content policy to the video pair matching stage, that is, only candidates exceeding a certain temporal coverage are considered duplicates. This not only ensures that partially overlapping videos are not misclassified, but is also fair when comparing videos with varying lengths, and supports policy-driven content management like manual review. To enforce such policies effectively, we propose DiF-SiM, a differential feature-enhanced similarity module in the video pair matching stage. DiF-SiM captures temporal relations between video frame through differential features, learned similarity metric and pre-training techniques leveraging unlabeled videos and unlabeled images.

To summarize, this paper makes the following contributions:

- We propose MLT-Dedup, which addresses the index capacity issue while reducing video repetition by integrating compact multi-level video representation and a carefully designed spatial-temporal matching model.
- In the video representation stage, we introduce ML-VE along with effective training strategies to generate video embeddings at both coarse and fine granularities. Coarse

embeddings are used for retrieval to improve candidate coverage and recall, while fine-grained embeddings enhance matching precision in downstream deduplication.

- In the pair-wise matching stage, we propose DiF-SiM, capable of locating duplicated temporal segments between video pairs and providing reliable similarity evidence to support policy-driven deduplication decisions.

Extensive experiments on a large-scale online platform show that MLT-Dedup reduces the online repetition rate by 91% at 90% precision, while achieving a 5× increase in indexing capacity under the same resource budget.

2 Method

2.1 Overall Framework

MLT-Dedup follows the standard three-stage video deduplication pipeline, consisting of video representation, candidate retrieval, and pairwise matching, as shown in Fig. 1. Given an input video, ML-VE first encodes a video as multi-level representations. Compact clip-level embeddings are indexed using a Hierarchical Navigable Small World (HNSW) graph-based approximate nearest neighbor index [32] to retrieve top-K candidate embeddings, which are subsequently aggregated to obtain candidate videos. Fine-grained frame-level embeddings are stored on disk and loaded on demand for precise matching. Retrieved candidate video embeddings are then examined by a fine-grained matching module, DiF-SiM, which localizes duplicated temporal overlap and outputs a similarity score. Finally, policy-driven deduplication decisions are made based on the proportion of overlapping time. The following subsections describe the design of ML-VE model, the clip-level embedding-based retrieval strategy, and the DiF-SiM matching module in detail.

2.2 Multi-Level Video Representations

The challenge of building effective large-scale video retrieval systems lies in maintaining high retrieval recall under a fixed resource budget. With limited memory and storage, the library can only store a small number of representations per video, which inevitably reduces the coverage of visual variations and leads to a drop in candidate recall. To address this trade-off, we propose ML-VE, a video

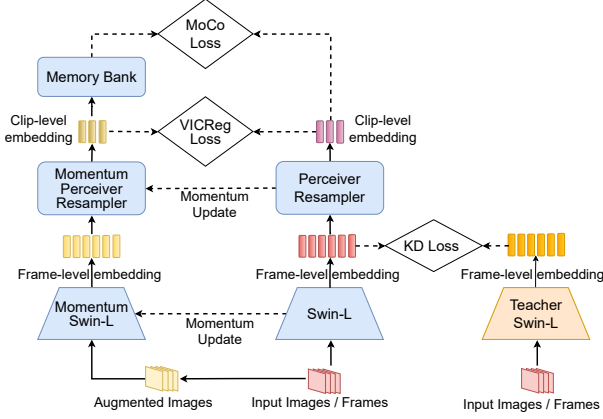


Figure 2: ML-VE architecture

encoder that generates multi-level video representations within a single forward pass. Compact clip-level embeddings are used for scalable retrieval, while fine-grained frame-level embeddings are retained for precise downstream matching. The effectiveness of these multi-level representations is achieved through a jointly designed model architecture and a set of targeted training strategies, such as hybrid loss formulation, inner-feature knowledge distillation and data-centric memory bank deduplication.

2.2.1 ML-VE Architecture. ML-VE is built upon a Swin-Large backbone [28] and trained under a MoCo-style momentum contrastive learning framework [12], as illustrated in Figure 2. For each video frame, the *online* (query) branch encodes the input with a Swin-Large encoder B_θ , producing frame-level visual tokens, and then applies a Perceiver Resampler R_θ to aggregate temporally distributed features into a fixed number of latent tokens. This yields compact clip-level token representations that are subsequently average-pooled into a D -dim global embedding for large-scale indexing and candidate retrieval. In parallel, we maintain a *momentum* (key) branch with the same architecture, parameterized by $\hat{\theta}$, which is updated as an exponential moving average (EMA) of the online parameters. During training, the momentum branch produces stable key embeddings that are enqueued into a memory bank (FIFO queue) to construct a large set of negative samples for contrastive learning. Our basic training objective consists of a VICReg [1] regularization loss and a MoCo [12] contrastive loss.

VICReg loss: We apply VICReg on paired embeddings from the two branches:

$$\mathcal{L}_{\text{VICReg}} = \lambda \mathcal{L}_{\text{inv}} + \mu \mathcal{L}_{\text{var}} + \nu \mathcal{L}_{\text{cov}}, \quad (1)$$

where λ, μ, ν are scalar weights. Let $Q \in \mathbb{R}^{B \times D}$ and $K \in \mathbb{R}^{B \times D}$ be the batch-stacked embeddings whose b -th rows are q_b^\top and k_b^\top ,

respectively. We define

$$\mathcal{L}_{\text{inv}} = \frac{1}{B} \sum_{b=1}^B \|q_b - k_b\|_2^2, \quad (2)$$

$$\mathcal{L}_{\text{var}} = \frac{1}{D} \sum_{d=1}^D \left[\max(0, \gamma - \sigma(Q_{:,d})) + \max(0, \gamma - \sigma(K_{:,d})) \right], \quad (3)$$

$$\mathcal{L}_{\text{cov}} = \frac{1}{D} \sum_{i \neq j} \left(\text{Cov}(Q)_{ij}^2 + \text{Cov}(K)_{ij}^2 \right), \quad (4)$$

where $\sigma(Q_{:,d})$ denotes the standard deviation of the d -th feature over the minibatch, $\gamma > 0$ is a margin, and $\text{Cov}(Q), \text{Cov}(K) \in \mathbb{R}^{D \times D}$ are covariance matrices computed on centered embeddings:

$$\text{Cov}(Q) = \frac{1}{B-1} \tilde{Q}^\top \tilde{Q}, \quad \text{Cov}(K) = \frac{1}{B-1} \tilde{K}^\top \tilde{K}, \quad (5)$$

with \tilde{Q} and \tilde{K} obtained by subtracting the per-dimension batch mean from Q and K .

MoCo loss: We maintain a memory bank (queue) Q of size K that stores negative keys $k^- \in \mathbb{R}^D$ produced by the momentum branch. With temperature τ and similarity function $\text{sim}(\cdot, \cdot)$, the MoCo (InfoNCE) loss is defined as

$$\mathcal{L}_{\text{MoCo}} = -\frac{1}{B} \sum_{b=1}^B \log \frac{\exp(\text{sim}(q_b, k_b)/\tau)}{\exp(\text{sim}(q_b, k_b)/\tau) + \sum_{k^- \in Q} \exp(\text{sim}(q_b, k^-)/\tau)}. \quad (6)$$

2.2.2 Hybrid Loss Formulation. To further enhance the discriminative power of the learned embeddings for retrieval and matching, we incorporate a supervised pairwise loss and a triplet loss on top of the self-supervised pre-training. This formulation combines the generalization ability of self-supervised representation learning with task-specific supervision, encouraging the learned embeddings to better distinguish near-duplicate and non-duplicate videos in retrieval and matching scenarios.

Pairwise Loss: We use a labeled pair dataset $\mathcal{D}_{\text{pair}} = \{(v_i, v_j, y_{ij})\}$ with $y_{ij} \in \{0, 1\}$ indicating whether (v_i, v_j) is a near-duplicate pair. Let $q_i, q_j \in \mathbb{R}^D$ be the pooled embeddings produced by the online branch for v_i and v_j . Using similarity $s_{ij} = \text{sim}(q_i, q_j)$ (consistent with MoCo), we optimize a logistic pairwise loss:

$$\mathcal{L}_{\text{match}} = -\frac{1}{|\mathcal{D}_{\text{pair}}|} \sum_{(i,j)} \left[y_{ij} \log \sigma(s_{ij}) + (1 - y_{ij}) \log(1 - \sigma(s_{ij})) \right], \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid function.

Triplet Loss: We form triplets $\mathcal{D}_{\text{trip}} = \{(v_a, v_p, v_n)\}$ from labels, where (v_a, v_p) is a positive (duplicate) pair and (v_a, v_n) is a negative pair. Let $q_a, q_p, q_n \in \mathbb{R}^D$ denote their online pooled embeddings. We use a margin-based triplet loss:

$$\mathcal{L}_{\text{tri}} = \frac{1}{|\mathcal{D}_{\text{trip}}|} \sum_{(a,p,n)} \max(0, \text{sim}(q_a, q_n) - \text{sim}(q_a, q_p) + m), \quad (8)$$

where m is the margin. This objective enforces that positive pairs are embedded closer than negative ones by a fixed margin.

2.2.3 Inner-Feature Knowledge Distillation. In practice, we observe that joint optimization with retrieval-oriented objectives biases ML-VE toward clip-level representations, while the quality of frame-level embeddings degrades, leading to suboptimal downstream matching performance. To explicitly enhance frame-level features, we introduce an inner-feature knowledge distillation (KD) strategy.

We adopt a Swin-Large teacher model trained to produce matching-friendly frame-level embeddings. The teacher model is trained on approximately 7M image samples using only a MoCo contrastive loss applied to the average-pooled token embedding, encouraging strong local visual semantics without clip-level aggregation bias. During ML-VE training, we apply feature-level distillation at multiple intermediate layers of the Swin-Large backbone. Let $f_\theta^{(l)}$ and $f_T^{(l)}$ denote the student and teacher feature maps at layer l . The distillation loss is defined as:

$$\mathcal{L}_{KD} = \sum_{l \in \mathcal{L}} \left\| \text{Norm}(f_\theta^{(l)}) - \text{Norm}(f_T^{(l)}) \right\|_2^2. \quad (9)$$

This inner-feature supervision improves the discriminability of frame-level embeddings for matching, without increasing indexing memory or affecting retrieval efficiency.

To summarize, the overall objective is:

$$\mathcal{L} = \mathcal{L}_{\text{MoCo}} + \alpha \mathcal{L}_{\text{VICReg}} + \beta \mathcal{L}_{\text{match}} + \rho \mathcal{L}_{\text{tri}} + \omega \mathcal{L}_{\text{KD}}, \quad (10)$$

where $\alpha, \beta, \rho, \omega$ balance VICReg, MoCo, pairwise, triplet and KD loss.

2.2.4 Data-Centric Memory Bank Deduplication. As commonly observed in large-scale contrastive learning, random sampling over massive datasets inevitably introduces duplicated or near-duplicated samples. Such redundancy in the training data reduces the number of unique negatives and limits the effectiveness of contrastive supervision. To mitigate this issue, we adopt a query-adaptive filtering strategy when computing the contrastive loss with the memory bank. For each query embedding, we compute cosine similarities against all key embeddings stored in the memory bank. Based on the similarity scores, we identify near-positive or highly similar samples and exclude them from the negative set. The contrastive loss is then computed only with respect to the remaining dissimilar keys, which serve as valid negatives for the given query. This data-centric, query-level filtering preserves the diversity of the memory bank while preventing false negatives from dominating the training signal.

2.3 Sparse Retrieval

We perform retrieval on sparse clip-level representations generated by ML-VE, rather than dense frame-level embeddings, to improve indexing efficiency and expand candidate recall coverage under fixed memory budgets.

Specifically, compact clip-level embeddings are indexed using an HNSW-based approximate nearest neighbor structure [19], enabling scalable retrieval with low memory overhead while allowing a larger portion of the video corpus to be retained in the index. Given a query video, its clip-level embeddings are independently queried against the HNSW index to retrieve nearest-neighbor embeddings. Each retrieved embedding corresponds to a specific clip from a candidate video. We then aggregate these clip-level matches into video-level similarity scores through a clip-to-video aggregation process. Retrieved results are first grouped by candidate video ID. If multiple clips from the same candidate video are retrieved by the same query clip, only the match with the highest similarity score is retained to avoid redundant contributions from temporally adjacent segments. Subsequently, for each candidate video, we collect its retained similarity scores across all query clips, discard

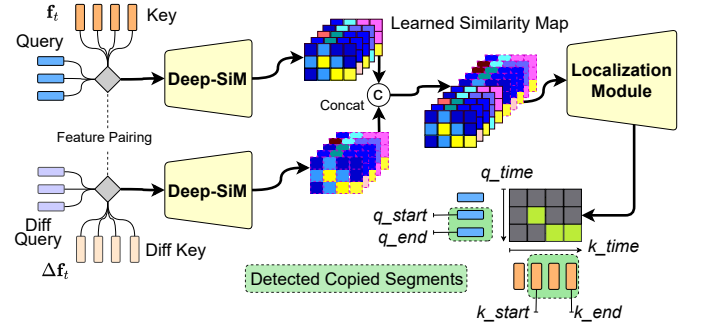


Figure 3: DiF-SiM Model Architecture: Static frame-level features from the ML-VE are used together with their differential features to produce similarity maps by Deep-SiM. They are then concatenated and fed into a localization module to predict the temporal boundaries of duplicated segments that dictate the timestamps q_start , q_end , k_start , and k_end

low-confidence matches below a similarity threshold (i.e., 0.4), and average the remaining scores to obtain a single video-level similarity score. Finally, candidate videos are ranked by their aggregated similarity scores, and the top- K videos are selected as retrieval results for downstream fine-grained matching.

2.4 Spatial-Temporal Matching

In this stage, we determine duplication based on the proportion of overlapping content in time and deduplication score: **a video is flagged as a duplicate only when the duplicated temporal coverage exceeds a predefined threshold, as well as having a deduplication score over the threshold.** This strategy avoids misclassifying partially overlapping videos, enables fair comparison across videos of different lengths, and provides explicit temporal localization for downstream auditing and manual review.

We observe that using clip-level embeddings from ML-VE directly for matching leads to degraded accuracy, especially for partially duplicated or temporally misaligned videos. Frame-level embeddings better preserve fine-grained visual and temporal details, enabling more accurate modeling of localized content reuse. Thus, the spatial-temporal matching stage performs fine-grained verification on frame-level representations. Only the embeddings of retrieved video pairs are loaded and processed in this stage, which allows the matching stage to remain compute-efficient.

To effectively leverage these fine-grained spatial-temporal matching signals, we propose DiF-SiM, a differential feature-enhanced similarity module for spatial-temporal video matching. DiF-SiM models frame-level correspondences between video pairs, localizes duplicated temporal segments, and produces similarity evidence that directly supports proportion-based deduplication policies. The architecture and matching strategy are detailed as follows.

2.4.1 DiF-SiM Model Architecture. As shown in Figure 3, frame-level embeddings from ML-VE are used as inputs of DiF-SiM. To model temporal correspondence, we also compute differential features by taking frame-wise differences of the embeddings, which highlight local temporal changes and are complementary to the

static appearance cues. Both sets of features are passed to a deep similarity module (Deep-Sim), which learns an enhanced similarity map between query and key feature pairs. The learned similarity maps are concatenated with cosine similarity maps and passed through a localization module to detect copied temporal video segments. Finally, DiF-SiM outputs the estimated overlap duration for both videos and a deduplication score ([q_start, q_end, k_start, k_end of, score]). Only when both the predicted overlap duration and the deduplication score exceed predefined thresholds could we consider the video pair to be duplicate.

2.4.2 Differential Features. In spatial-temporal video matching, directly comparing query and candidate videos via independent frame-to-frame similarity is insufficient for accurately modeling partial duplication and temporal misalignment. Such static pairwise comparisons ignore temporal transitions within each video, making it difficult to distinguish true content reuse from incidental visual similarity and to robustly localize duplicated segments.

To address this limitation, we leverage both conventional static frame features and additional *differential features* that capture video dynamics within each video. Given a sequence of frame-level embeddings $\mathbf{f}_t \in \mathbb{R}^D, t = 1, \dots, T$, extracted by ML-VE, we compute normalized differences between consecutive frames:

$$\Delta \mathbf{f}_t = \text{Norm}(\mathbf{f}_{t+1} - \mathbf{f}_t), \quad (11)$$

where $\text{Norm}(\cdot)$ denotes ℓ_2 normalization. To maintain a consistent feature length, we apply wrap-around at the last frame: $\Delta \mathbf{f}_T = \text{Norm}(\mathbf{f}_1 - \mathbf{f}_T)$.

The resulting differential features characterize local temporal transitions and motion patterns within a video, explicitly encode motion cues between adjacent frames, and complement static appearance information encoded in the original static embeddings. As illustrated in Appendix A, similarity maps derived from differential features exhibit clearer temporal alignment, especially under partial duplication and temporal shifts.

2.4.3 Deep-Sim: Deep Similarity Module. In the spatial-temporal matching stage, similarity maps are computed to reflect temporal alignment. However, directly computing similarity using a fixed cosine metric between feature vectors is often insufficient, because it compresses rich relationships into a single scalar value and lacks the flexibility to integrate heterogeneous cues. To handle this issue, we introduce a *Deep Similarity Module* (Deep-Sim), which learns a parametric similarity metric that maps a pair of features to a high-dimensional similarity representation. Instead of relying on a hand-crafted metric, Deep-Sim models similarity as a learnable function, enabling richer semantic alignment between video elements.

We pre-train Deep-Sim in a self-supervised manner using unlabeled image data to endow it with general-purpose similarity modeling capability. Specifically, we attach an auxiliary linear head \mathcal{M} to the Deep-Sim network \mathcal{F} . Deep-Sim is optimized by a binary cross-entropy loss to distinguish matching and non-matching pairs. This pre-training stage encourages Deep-Sim to learn discriminative similarity patterns beyond simple feature distance. After pre-training, we discard the auxiliary head \mathcal{M} and transfer the learned weights of \mathcal{F} into the DiF-SiM module. In the full system, Deep-Sim is further fine-tuned jointly with the downstream localization module, allowing similarity estimation to adapt to video-level temporal

matching objectives. We detail the Deep-Sim structure in Appendix B.

Deep-Sim is applied to both static frame features and differential features, producing two enhanced similarity maps that capture complementary static and temporal correspondences. In parallel, we compute a conventional cosine similarity map on both static frame features and differential features. These four similarity maps are concatenated along the channel dimension and fed into the localization module to infer duplicated temporal segments.

2.4.4 Localization Module. Our localization module follows previous works [13, 31] and use the YOLOX-s [10] detector with width= 0.33, depth= 0.33, with an input shape of 640×640 . To accommodate similarity-map inputs, we modify the shape of the first convolutional layer accordingly. The localization module performs detection on a two-dimensional time plane—each bounding-box corresponds to a "time-time" rectangular window (horizontal axis for query time, vertical axis for candidate time), indicating that two time intervals may match with each other. The localization module outputs the estimated overlap boundaries for both videos, together with a deduplication confidence score.

2.4.5 Self-Supervised pre-training with Dual-Modality Data Augmentation. To better initialize the model and reduce reliance on the number of paired labels, we pre-train DiF-SiM via self-supervised learning (SSL) [4, 6, 11]. In raw unlabeled video collections, true duplicated pairs are extremely sparse. Manual data annotation for the video copy localization task is difficult and expensive. To address this, we leverage two modalities of unlabeled data, images and videos, to improve model performance via self-supervised learning (SSL) [4, 6, 11] during model pre-training.

Video SSL. We leverage unlabeled video data to pre-train the entire DiF-SiM under realistic copy-and-edit scenarios. Inspired by prior work [31], we design three lightweight video-level augmentation strategies, as illustrated in Appendix C.

- *Self-insertion* generates duplicate pairs by randomly selecting and reinserting segments within the same video, modeling rapid edits and temporal shifts.
- *Single-insertion* inserts segments from one video into another, combined with visual and temporal perturbations, to simulate common cross-video duplication such as reuploads with trimming or speed changes.
- *Double-insertion* inserts segments from a source video into two different background videos, producing challenging cases with high semantic similarity but different contexts.

Image SSL. We leverage unlabeled image data to pre-train the Deep-Sim module. For each mini-batch of unlabeled images, we apply a random augmentation pipeline (i.e., cropping, flipping, color jittering) to each image and feeds the augmented image into ML-VE to obtain an embedding. It forms a positive pair, labeled as 1, that comes from the same original image. A negative pair comes from a different image within this batch, labeled as 0.

Together, these transformations expose the model to diverse duplication patterns and strengthen its ability to localize copied regions under temporal misalignment and visual variations.

3 Experiments

3.1 Datasets

ML-VE model. ML-VE is trained on a comprehensive dataset combining 24 million in-house videos and 4.6 billion open-source images from LAION-2B [37]. Most of the in-house videos are unlabeled, which are used for self-supervised training. A small portion of labeled pairwise and triplet data are used for supervised learning. The test set includes 1k in-house query videos, each having 20 to 100 candidates.

DiF-SiM model. In production, DiF-SiM is trained on 0.5M human-annotated in-house video clip pairs, together with 2M image samples from Conceptual Captions [38], and 3 million video SSL pre-training data from the InternVid dataset [41], for self-supervised pre-training. All videos are first processed by the ML-EV model to extract 256-d frame-level embeddings, which serve as inputs to DiF-SiM. Labels are discarded for the SSL data.

3.2 Implementation Details

For both ML-VE and DiF-SiM, training utilizes 16 NVIDIA A100 GPUs. We use AdamW optimizer [30] with learning rate 0.0002, betas of [0.9, 0.999], weight decay of 0.01, and epsilon of $1e^{-8}$. We utilize a cosine annealing learning rate scheduler with 100 warm-up steps. For each video clip, we regard 16 frames as a clip. ML-VE outputs 768-dimensional clip-level embeddings for downstream retrieval, and 256-dimensional frame-level embeddings for DiF-SiM matching. DiF-SiM finally outputs 640×640 map indicating overlap times and duplication scores.

3.3 Evaluation Results

3.3.1 Overall Online Performances. To evaluate real-world impact, we conducted online A/B tests by incrementally deploying each component. The experiments ran for several weeks, with 10% of traffic allocated to each experimental group. We evaluate online performance using the repetition rate, defined as the proportion of duplicated video pairs among all distinct video pairs formed from videos recommended to the same user, where duplication is determined via human annotation. Statistical significance is assessed using a two-sample z-test for proportions. The baseline system employs Swin-Large [28] to extract frame-level video representations and uses a non-parametric cosine matcher for fine-grained similarity computation by directly comparing frame embeddings. All online results are reported at a fixed precision of 90%.

As shown in Table 1, the proposed MLT-Dedup system achieves substantial reductions in content repetition. Replacing the baseline video representation model with ML-VE alone reduces the repetition rate by 72%, demonstrating that sparse clip-level retrieval dramatically improves candidate coverage and effectively mitigates missed duplicates at the retrieval stage. When only upgrading the matching model, replacing the cosine matcher with DiF-SiM yields a 24% reduction in repetition rate. This result indicates that learned, parametric spatial-temporal verification can filter out a significant portion of false positives that cannot be resolved by static, frame-wise similarity measures. By jointly integrating ML-VE and DiF-SiM, MLT-Dedup achieves a total repetition rate reduction of **91%**. The

Table 1: Online Deduplication Performance the overall pipeline at 90% precision. We show the reduction in repetition rates due to the addition of our ML-VE and DiF-SiM matcher to our baseline. “Swin-L” denotes “Swin-Large”

Method	Repetition Rate	Index Size
Swin-L + Cosine Matcher (Baseline)	–	1
ML-VE + Cosine Matcher	-72%	×5
Swin-L + DiF-SiM	-24%	×1
ML-VE + DiF-SiM (MLT-Dedup)	-91%	×5

Table 2: Comparison between different video representation models for recall evaluated on our in-house data. “dim” denotes “dimension”

Model	Frames/Emb	Dim	Recall@5
Webssl-Dino-7b-Full-8b-518	1	4096	82.60
Dinov2-Large	1	1024	85.49
GME-Qwen2-VL-2B	1	3584	83.67
MM_Embed	1	4096	82.76
VLM2Vec	1	4096	80.68
E5-V	1	4096	79.91
ML-VE (ours)	16	768	87.59

gains arise from stage-wise synergy, where ML-VE enhances candidate retrieval and DiF-SiM refines spatial-temporal matching. In addition, the sparse clip-level representations produced by ML-VE significantly improve indexing efficiency. With a fixed amount of index resources, MLT-Dedup stores up to **5×** more content in the retrieval index, enabling a longer index Time To Live (TTL) without sacrificing deduplication accuracy.

Table 3: Comparison of the performance on the VCSL test set of our DiF-SiM model against open-source baselines. All methods utilize the same set of frame level embeddings extracted by ISC encoder

Method	F-score	Precision	Recall
HV	51.73	36.82	86.94
TN	64.43	66.50	62.49
DP	54.53	60.63	49.56
DTW	50.23	56.67	45.10
SPD	62.97	56.27	71.47
TransVCL	66.51	67.46	65.59
RTR	69.22	66.77	71.85
RTR + pre-training	70.73	68.19	73.46
DiF-SiM (ours)	74.31	71.21	77.68

3.3.2 Evaluation of ML-VE. We compare our proposed model with open-source video representation models (Webssl-Dino-7b-Full-8b-518 [9], Dinov2-Large [34], GME-Qwen2-VL-2B [43], MM_Embed [25], E5-V [17]) under Recall@5, which measures the proportion of ground-truth relevant videos that appear in the top-5 retrieved results. Each method is evaluated on the same 1k clusters. As shown

in Table 2, despite using the coarse temporal granularity of one embedding per 16 frames, ML-VE achieves the highest Recall@5 among all compared models. In contrast, most baseline models rely on frame-level representations (one frame per embedding) with significantly higher embedding dimensionality. This suggests that ML-VE improves recall not by denser representations, but by leveraging coarse aggregation to expand the retrievable candidate set while retaining sufficient semantic discriminability.

3.3.3 Evaluation of DiF-SiM. On the in-house test set, DiF-SiM achieves strong performance across multiple metrics:

- F1: 86.79
- AP: 94.01
- R@P90: 83.32

which is achieved not only by DiF-SiM’s DeepSim, differential features, and SSL pre-training, but also through the high-quality frame-level embeddings provided by ML-VE.

To extend the comparison to open-source video-deduplication models, we follow the protocol from VCSL [14] and directly use their provided 256-d ISC embeddings as model inputs. Training leverages InternVid [41], Conceptual Captions [38], and the full VCSL training split, and evaluation is conducted on 55K VCSL test pairs with a balanced distribution of positives and negatives. Compared methods include Hough Voting (*HV*) [8], Temporal Networks (*TN*) [40], Dynamic Programming (*DP*) [5], Dynamic Time Warping (*DTW*) [2], Similarity Pattern Detection (*SPD*) [16], TransVCL [13] and Regional Token Representation (*RTR*) [31]. We adopt the F-score, which is the harmonic mean between the segment-level recall and precision defined by VCSL [17] as the metric to evaluate the effectiveness of temporal localization of copied segments.

As shown in Table 3, our model achieves an F-score of 74.31%, outperforming the previous state of the art (*RTR + pre-training*) by **3.58**. These results demonstrate that DiF-SiM delivers strong and competitive performance on the public VCSL benchmark, effectively matching and localizing copied video segments under standardized evaluation protocols. We also compare results across all categories of the VCSL testset in Appendix D. Visualizations for video copy localization results compared to RTR are shown in Appendix E.

3.3.4 Ablations of ML-VE. We conduct ablation studies on ML-VE by adding a hybrid loss and memory bank deduplication on our baseline backbone, which is a Swin-Large backbone with MoCo-style contrastive loss, regularization loss and inner-feature knowledge distillation. As shown in Table 4, introducing a hybrid loss that combines pairwise and triplet objectives improves Recall@5 from 82.46 to 85.72, as it explicitly enforces supervised learning and relative similarity ordering that better matches nearest-neighbor retrieval behavior. Enabling memory bank deduplication helps to mitigate redundant positives and enhance discriminative retrieval under a fixed memory budget, leading to further improvements in Recall@5.

We further evaluate the impact of inner-feature knowledge distillation (KD) in Table 5. Without KD, ML-VE suffers a clear performance drop in matching-related metrics, especially under strict operating points (i.e. P@R90, P@R95, R@P90, and R@P95). After introducing the KD loss, ML-VE achieves consistent improvements across all metrics, indicating that KD substantially strengthens the quality of frame-level representations used for downstream

Table 4: The effectiveness of hybrid loss and memory bank deduplication on ML-VE

Recall@5	Hybrid Loss	Memory Bank Deduplication
82.46	×	×
85.72	✓	×
87.59	✓	✓

Table 5: The effectiveness of inner-feature knowledge distillation (KD) on ML-VE.

Model	AP	F1	P@R90	P@R95	R@P90	R@P95
ML-VE w/o KD	91.31	83.69	66.09	47.80	77.38	74.68
ML-VE	94.01	86.79	78.18	62.05	83.32	79.34
Swin-L (teacher)	94.29	87.05	79.35	63.24	84.05	78.66

Table 6: Effectiveness of DiF-SiM components on both the VCSL test set and our in-house test set, “ f_t ”, “ Δf_t ”, “iPT”, “vPT” denote “static features”, “differential features”, “image SSL pre-training” and “video SSL pre-training”, respectively.

Method				In-House test set			VCSL test set
Deep f_t	Deep Δf_t	iPT	vPT	F1	AP	R@P90	F-score
×	×	×	×	83.87	91.49	77.14	70.20
✓	×	×	×	84.72	92.63	79.86	72.14
✓	✓	×	×	86.07	93.03	81.22	73.10
✓	✓	✓	×	86.25	93.55	81.89	73.52
✓	✓	✓	✓	86.79	94.01	83.32	74.31

matching, and closes the gap between matching performance to the teacher model.

3.3.5 Ablations of DiF-SiM. We validate the effectiveness of differential features, Deep-Sim, and SSL pre-training strategy on both the open-source VCSL and in-house test sets. For fair comparison, we use 256-d ISC embeddings and train models on VCSL, while 256-d ML-EV frame-level embeddings and corresponding in-house training data are used for business scenarios. As shown in Table 6, the first row corresponds to the baseline that applies cosine similarity on static frame-level appearance features only. Compared to this baseline, Deep-Sim contributes the dominant performance gain by introducing parametric similarity modeling, which significantly improves all metrics by allowing richer correspondence patterns to be learned beyond a fixed similarity metric. Incorporating differential features on top of Deep-Sim further boosts high-precision recall, indicating their effectiveness in capturing temporal transitions and motion consistency that are critical for accurate duplication localization. Image- and video-level self-supervised pre-training progressively stabilize and adapt the learned similarity space to realistic appearance and editing variations.

We further validate the generalization ability of Deep-Sim and differential features. Taking TransVCL as the backbone, we replace the original self-attention and cross-attention modules with Deep-Sim, using either static frame features or differential features as

Table 7: Performance after replacing TransVCL’s attention-based similarity mapping with Deep-Sim on VCSL test set, “ \mathbf{f}_t ” and “ $\Delta\mathbf{f}_t$ ” denote “static feature” and denotes “differential feature”, respectively

Similarity Mapping Method	Similarity Map Dim	Feature	F-score	Precision	Recall
TransVCL Original Attention	1	\mathbf{f}_t	66.51	67.46	65.59
Deep-Sim	8	\mathbf{f}_t	70.65	68.53	72.90
Deep-Sim	16	\mathbf{f}_t	71.59	72.75	70.48
Deep-Sim	64	\mathbf{f}_t	72.14	72.21	72.07
Deep-Sim	8	$\mathbf{f}_t + \Delta\mathbf{f}_t$	72.04	69.95	74.98
Deep-Sim	16	$\mathbf{f}_t + \Delta\mathbf{f}_t$	72.62	70.20	75.20
Deep-Sim	64	$\mathbf{f}_t + \Delta\mathbf{f}_t$	73.10	73.57	72.62

input. We illustrate results in Table 7. Comparing the second and third row, we find that using only static frame features with Deep-Sim already brings consistent gains over the TransVCL baseline. Incorporating differential features further boosts performance across all tested dimensions. At the same dimension, differential versions consistently outperform static ones (i.e., +1.39 F-score at 8d, +1.03 at 16d, +0.96 at 64d). Within each feature type, increasing the frame dimension has a clear positive impact on overall performance. This indicates that higher-dimensional representations allow Deep-Sim to capture more nuanced similarity patterns without overfitting, while the differential input provides complementary information that scales well with dimension.

4 Related Works

4.1 Video Deduplication Pipeline

Video deduplication pipelines have been extensively studied in multimedia and computer vision [21, 26, 35, 39], driven by applications including copyright protection, web video management, recommendation, and large-scale duplicate filtering. With growing video scale, deep learning-based representations and metric learning are applied in the pipeline for better robustness. A prominent recent paradigm adopts a three-stage pipeline: video representation, candidate retrieval, and fine-grained matching [3, 23, 39]. However, existing pipelines still suffer from limited index capacity in retrieval and imprecise alignment under partial edits and speed change. To tackle this issue, in this paper, we adhere to the classic three-stage pipeline while introducing multi-level embeddings in video representation and spatial-temporal matching. Our proposed pipeline expands index capacity with sustained high recall, and yielding higher precision and significantly lower duplication rates.

4.2 Video Representation Model

Recent progress in visual representation learning has substantially improved video feature extraction, driven by strong backbones such as ViT [7] and Swin Transformer [28], as well as self-supervised approaches including MoCo [12, 34]. More recently, large-scale embedding models [17, 18, 24, 25, 33, 43] further enhance fine-grained frame-level representations. In parallel, multi-frame aggregation

architectures such as Video Swin Transformer [29] and Perceiver [15] enable compact clip-level representations. While single-frame features achieve strong retrieval performance, they require substantially more indexing capacity, whereas clip-level features are more resource-efficient but tend to yield lower recall and significantly degrade the effectiveness of downstream fine-grained matching modules. Motivated by these advances and real-world deduplication requirements, we design a specialized video representation model that jointly outputs fine-grained frame-level and compact clip-level embeddings. This dual-level representation significantly reduces indexing cost and improves retrieval efficiency, while preserving precise spatial-temporal cues for accurate matching in large-scale online video deduplication.

4.3 Matching methods in Deduplication Pipeline

Fine-grained matching plays a pivotal role in video deduplication pipeline, particularly in the third stage of coarse-to-fine architectures, where candidate videos are precisely verified and aligned to identify overlapping segments. Early matching methods (i.e., Hough Voting [8], DTW [2], DP [5], TN [40]) relied on fixed formulas (cosine / Euclidean) to build similarity maps, followed by rule-based temporal alignment (voting, path optimization). These methods are rigid and non-adaptive, failing to exploit end-to-end optimization, limiting overall performance in complex, large-scale scenarios. Recent works (i.e., SPD [16], TransVCL [13], RTR [31]) adopt learned similarity modules, achieving better robustness on partial copies. However, they still use static frame features and ignore differential information, wasting the natural dynamic information in video sequences. To address this, our method introduces a matching module that explicitly incorporates frame-to-frame differentials into the learned similarity computation for dynamic-aware alignment. By modeling motion residuals as auxiliary tokens, we enhance copy localization under temporal distortions, effectively exploiting video dynamics for more discriminative maps.

5 Conclusion

This paper presents **MLT-Dedup**, which combines **Multi-Level** video representations with **spatial-Temporal** matching for large-scale, efficient, online video deduplication. We introduce **ML-VE** to improve candidate coverage under a minimal indexing budget through leveraging task-specific video representations. Two scales of embeddings are generated simultaneously: compact segment-level embeddings for scalable retrieval and frame-level embeddings for fine-grained copy localization. For precise video duplication verification, we introduce **DiF-SiM**, a **Differential Feature-enhanced Similarity Module**, to accurately score recalled pairs. DiF-SiM performs learned similarity modeling and integrates differential features to better capture temporal dynamics within video pairs, and is further strengthened through self-supervised pre-training techniques. Extensive experiments on the public VCSL benchmark and a large in-house dataset show consistent improvements over strong baselines, demonstrating the effectiveness of our system for real-world deduplication. We deploy **MLT-Dedup** online to serve full production traffic on a large-scale short-video platform and observe significant reductions in video copy rates.

References

- [1] Adrien Bardes, Jean Ponce, and Yann LeCun. 2022. VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning. In *ICLR*.
- [2] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD Workshop*. <https://api.semanticscholar.org/CorpusID:929893>
- [3] Alexander Black, Simon Jenni, Tu Bui, Md Mehrab Tanjim, Stefano Petrangeli, Ritwik Sinha, Viswanathan Swaminathan, and John Collomosse. 2023. Vader: Video alignment differencing and retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22357–22367.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PmLR, 1597–1607.
- [5] Chien-Li Chou, Hua-Tsung Chen, and Suh-Yin Lee. 2015. Pattern-Based Near-Duplicate Video Retrieval and Localization on Web-Scale Videos. *IEEE Transactions on Multimedia* 17, 3 (2015), 382–395. doi:10.1109/TMM.2015.2391674
- [6] Virginia R. de Sa. 1993. Learning Classification with Unlabeled Data. In *Neural Information Processing Systems*. <https://api.semanticscholar.org/CorpusID:9890353>
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929 [cs.CV] <https://arxiv.org/abs/2010.11929>
- [8] Matthijs Douze, Hervé Jégou, Cordelia Schmid, and Patrick Pérez. 2010. Compact Video Description for Copy Detection with Precise Temporal Alignment. In *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 522–535.
- [9] David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, and Saining Xie. 2025. Scaling Language-Free Visual Representation Learning. arXiv:2504.01017 [cs.CV] <https://arxiv.org/abs/2504.01017>
- [10] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: Exceeding YOLO Series in 2021. arXiv:2107.08430 [cs.CV] <https://arxiv.org/abs/2107.08430>
- [11] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. 2024. A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. arXiv:2301.05712 [cs.LG] <https://arxiv.org/abs/2301.05712>
- [12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. arXiv:1911.05722 [cs.CV] <https://arxiv.org/abs/1911.05722>
- [13] Sifeng He, Yue He, Minlong Lu, Chen Jiang, Xudong Yang, Feng Qian, Xiaobo Zhang, Lei Yang, and Jiandong Zhang. 2022. TransVCL: Attention-enhanced Video Copy Localization Network with Flexible Supervision. arXiv:2211.13090 [cs.CV] <https://arxiv.org/abs/2211.13090>
- [14] Sifeng He, Xudong Yang, Chen Jiang, et al. 2022. A Large-scale Comprehensive Dataset and Copy-overlap Aware Evaluation Protocol for Segment-level Video Copy Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21086–21095.
- [15] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. 2021. Perceiver: General Perception with Iterative Attention. arXiv:2103.03206 [cs.CV] <https://arxiv.org/abs/2103.03206>
- [16] Chen Jiang, Kaiming Huang, Sifeng He, Xudong Yang, Wei Zhang, Xiaobo Zhang, Yuan Cheng, Lei Yang, Qing Wang, Furong Xu, Tan Pan, and Wei Chu. 2021. Learning Segment Similarity and Alignment in Large-Scale Content Based Video Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, 1618–1626. doi:10.1145/3474085.3475301
- [17] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. 2024. E5-V: Universal Embeddings with Multimodal Large Language Models. arXiv:2407.12580 [cs.CL] <https://arxiv.org/abs/2407.12580>
- [18] Ziyang Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2025. VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks. arXiv:2410.05160 [cs.CV] <https://arxiv.org/abs/2410.05160>
- [19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. arXiv:1702.08734 [cs.CV] <https://arxiv.org/abs/1702.08734>
- [20] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. Finding near-duplicate videos in large-scale collections. In *Video Verification in the Fake News Era*. Springer, 91–126.
- [21] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. 2016. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *International conference on multimedia modeling*. Springer, 251–263.
- [22] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. 2017. Near-Duplicate Video Retrieval With Deep Metric Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.
- [23] Giorgos Kordopatis-Zilos, Christos Tzilepis, Symeon Papadopoulos, Ioannis Kompatsiaris, and Ioannis Patras. 2022. DnS: Distill-and-select for efficient and accurate video indexing and retrieval. *International Journal of Computer Vision* 130, 10 (2022), 2385–2407.
- [24] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. arXiv:2405.17428 [cs.CL] <https://arxiv.org/abs/2405.17428>
- [25] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. 2025. MM-Embed: Universal Multimodal Retrieval with Multimodal LLMs. arXiv:2411.02571 [cs.CL] <https://arxiv.org/abs/2411.02571>
- [26] Jiajun Liu, Zi Huang, Hongyun Cai, Heng Tao Shen, Chong Wah Ngo, and Wei Wang. 2013. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 1–23.
- [27] Yao Liu, Sam Blasiak, Weijun Xiao, Zhenhua Li, and Songqing Chen. 2015. A Quantitative Study of Video Duplicate Levels in YouTube. In *International Conference on Passive and Active Network Measurement*. Springer, 235–248.
- [28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv:2103.14030 [cs.CV] <https://arxiv.org/abs/2103.14030>
- [29] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2021. Video Swin Transformer. arXiv:2106.13230 [cs.CV] <https://arxiv.org/abs/2106.13230>
- [30] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs.LG] <https://arxiv.org/abs/1711.05101>
- [31] Minlong Lu, Yichen Lu, Siwei Nie, Xudong Yang, and Xiaobo Zhang. 2025. Self-supervised Video Copy Localization with Regional Token Representation. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 18–35.
- [32] Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arXiv:1603.09320 [cs.DS] <https://arxiv.org/abs/1603.09320>
- [33] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative Representational Instruction Tuning. arXiv:2402.09906 [cs.CL] <https://arxiv.org/abs/2402.09906>
- [34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. arXiv:2304.07193 [cs.CV] <https://arxiv.org/abs/2304.07193>
- [35] Dhanashree A Phalke, Sunita Jahirabaddkar, and SPPU Pune. 2018. A systematic review of near duplicate video retrieval techniques. *International Journal of Pure and Applied Mathematics* 118, 24 (2018), 1–11.
- [36] Tiago Rodrigues, Fabrício Benevenuto, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. 2010. Equal but different: a contextual analysis of duplicated videos on YouTube. *Journal of the Brazilian Computer Society* 16, 3 (2010), 201–214.
- [37] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems* 35 (2022), 25278–25294.
- [38] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of ACL*.
- [39] Ling Shen, Richang Hong, and Yanbin Hao. 2020. Advance on large scale near-duplicate video retrieval. *Frontiers of Computer Science* 14, 5 (2020), 145702.
- [40] Hung-Khoon Tan, Chong-Wah Ngo, Richard Hong, and Tat-Seng Chua. 2009. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of the 17th ACM International Conference on Multimedia (Beijing, China) (MM '09)*. Association for Computing Machinery, New York, NY, USA, 145–154. doi:10.1145/1631272.1631295
- [41] Yi Wang, Yanan He, Yizhuo Li, Kunchang Li, Jiahuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Conghui He, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. 2024. InternVid: A Large-scale Video-Text Dataset for Multimodal Understanding and Generation. arXiv:2307.06942 [cs.CV] <https://arxiv.org/abs/2307.06942>
- [42] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. 2007. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th ACM International Conference on Multimedia (Augsburg, Germany) (MM '07)*. Association for Computing Machinery, New York, NY, USA, 218–227. doi:10.1145/1291233.1291280
- [43] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2025. GME: Improving Universal Multimodal Retrieval by Multimodal LLMs. arXiv:2412.16855 [cs.CL] <https://arxiv.org/abs/2412.16855>

Appendix

A Differential Features vs Static Features

An example of a copied video pair is shown in Figure 4. We visualize the normalized similarity map formed from normal features (left) and differential features (right). Differential features exhibit a significantly cleaner pattern, more effectively highlighting potential areas of video duplication than the regular features.

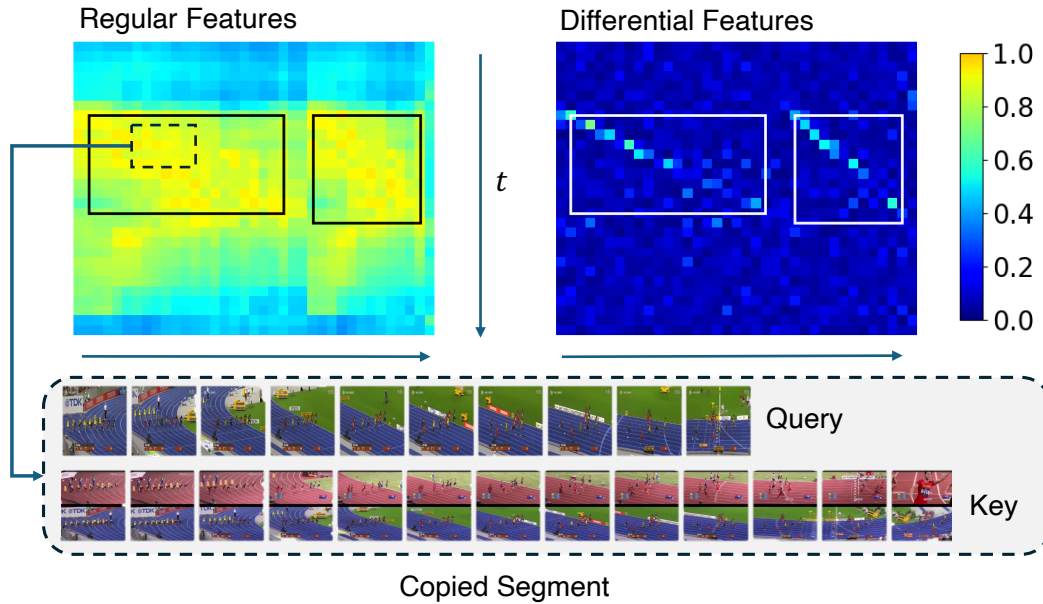


Figure 4: Similarity Map from Differential Features Compared to Static Features

B Deep-Sim Model Architecture

The architecture of the Deep-Sim module is shown in Figure 5. The input query and candidate features are concatenated and passed through a series of residual linear layers (Res-Block), followed by a multi-layer perceptron (MLP). The module learns an enhanced similarity for each query-key pair, achieving strong semantics compared to direct cosine similarity calculation. During SSL pre-training, an additional MLP is added to enhance model training by using transformed image pairs in place of the query and key.

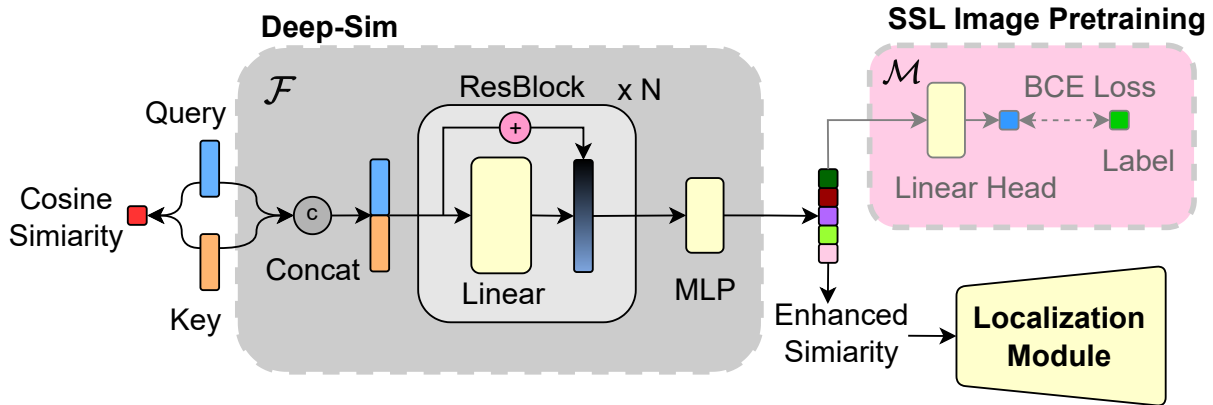


Figure 5: Deep-Sim Model Architecture

C DiF-SiM Self-Supervised Pre-training with Dual-Modality Data Augmentation

We visualize the self-supervised augmentations used in our video pre-training for DiF-SiM. Figure 6a, 6b, and 6c show self-insertion transform, single-insertion transform, and double-insertion transform, respectively.

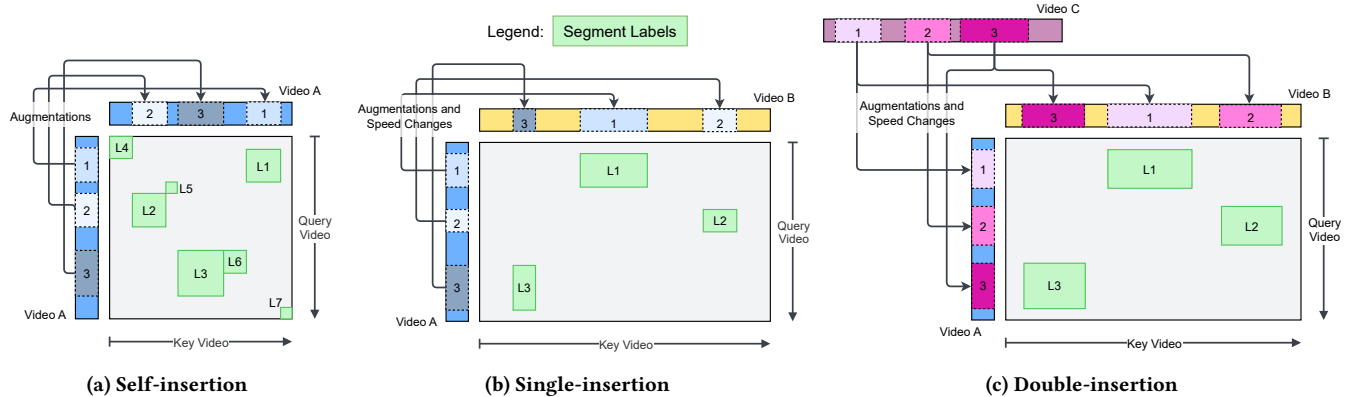


Figure 6: Illustrations of the augmentations utilized for self-supervised training. (a) Self-insertion. Both the query and key videos are created from a single video A. Random segments from the query are replaced into the key, creating segment labels L1-L3. Since query and key are the same video, the remaining un-replaced key segments also form copied segments, shown as L4-L7. **(b) Single-insertion.** The query and key videos are created from two videos A and B. Random segments from the query are inserted into the key to form segment labels L1-L3. **(c) Double-insertion.** The query and key are derived from two videos A and B. Random segments from a third video C are inserted into both the query and key to form segment labels L1-L3

D DiF-SiM Video Copy Localization Performance Across Categories

Table 8: Performance of our model across all categories of videos in the VCSL test set, compared to previous TransVCL and RTR results. The best performance is in bold. We see that our model exhibit top results in 10/11 categories.

METHODS	VARIETY SHOW	GAMES	MUSIC VIDEOS	NEWS	SPORTS	DAILY LIFE	TVSERIES	KICHIKU	ADVERT-ISEMENT	ANIM-ATION	MOVIES
TRANSVCL	91.25	85.25	81.44	74.02	66.81	56.61	91.29	56.67	68.18	90.84	71.80
RTR	86.62	96.22	88.94	87.30	66.65	62.19	95.35	63.62	72.40	96.42	75.42
DiF-SiM (OURS)	79.81	97.88	91.96	92.91	75.49	69.85	95.97	68.68	81.26	97.29	77.90

We evaluate DiF-SiM across diverse video categories of the VCSL dataset [14]: variety shows, games, music videos, news, sports, daily life, TV series, Kichiku, advertisements, animation, and movies. The results, as summarized in Table 8, are compared with the previous state-of-the-art methods, RTR [31] and TransVCL [14]. We compare F-scores across all 11 categories and observe that our model achieves the highest performance in most of the categories.

E Video Localization Result Visualizations

We show some visualization examples on the VCSL video localization test set by our proposed DiF-SiM matching model. We visualize and compare to the predictions made from RTR [31]. Visualizations shown in Figures 7, 8, and 9 demonstrates that our differential features and DiF-SiM effectively resolve ambiguous high-similarity cases where RTR fails, leading to more accurate temporal copy localization and substantially higher F1 score.

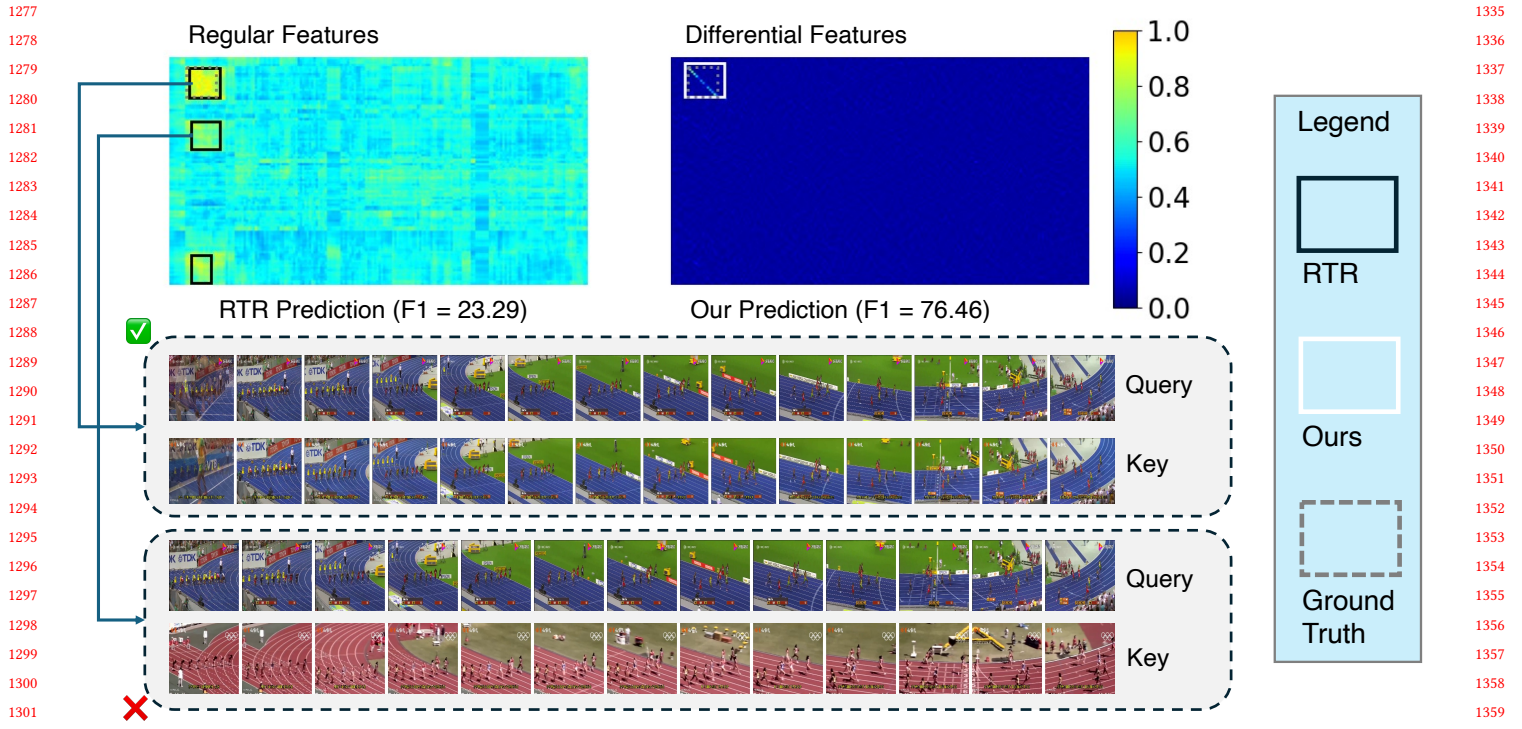


Figure 7: An example of video copy localization predictions made by the RTR model (left) and our model (right). We show the RTR prediction using black bounding boxes and our predictions using white bounding boxes. The ground-truth label is shown using gray dotted bounding boxes. The cosine similarity map is visualized on the left and the differential map visualized on the right. Below, we show the video frames for the correctly predicted section (top) and the incorrectly predicted section by RTR (bottom). We see that both of these sections exhibit high similarity, but the latter pair are not a duplicated pair. Our differential features and DiF-SiM modules help to more clearly differentiate between such hard cases to make more accurate predictions, achieving much higher F1 score than RTR.

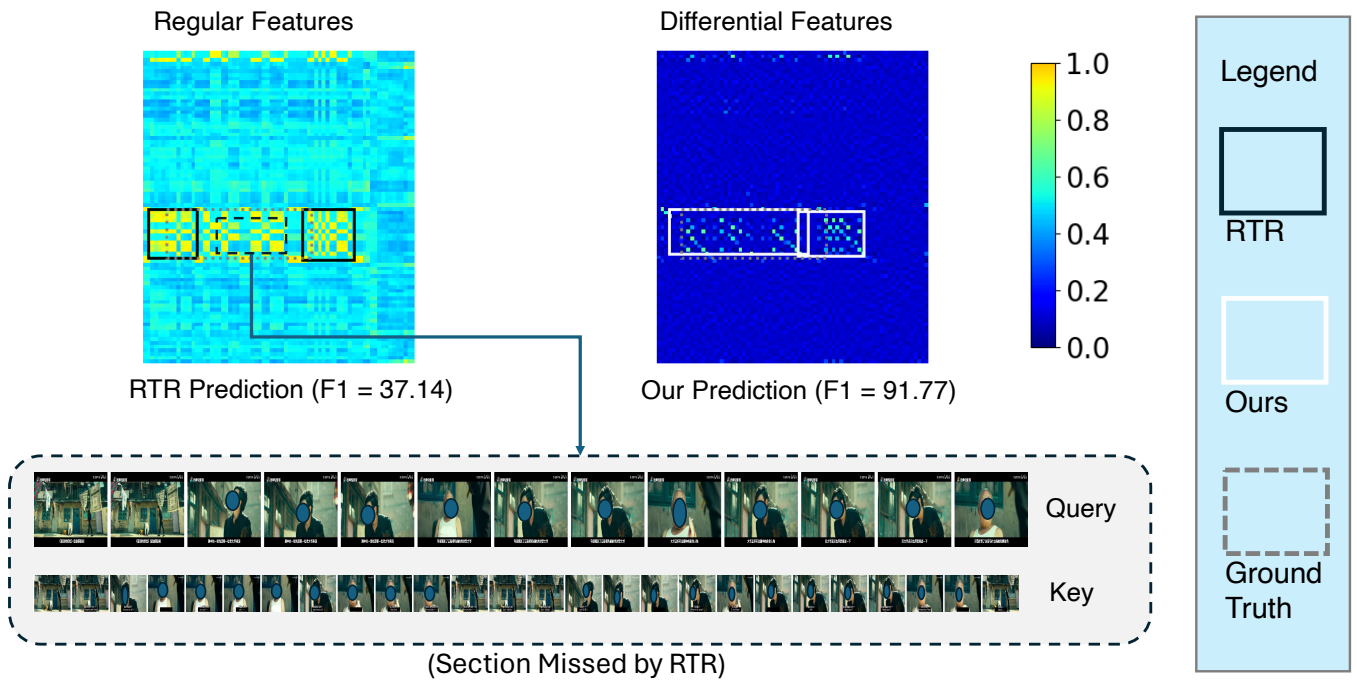


Figure 8: An example of video copy localization predictions made by the RTR model (left) and our model (right). We show the RTR prediction using black bounding boxes and our predictions using white bounding boxes. The ground-truth label is shown using gray dotted bounding boxes. The cosine similarity map is visualized on the left and the differential map visualized on the right. Below, we show the video frames for the section of duplicated video which is missed by RTR. By studying these frames, we see that the key video has undergone intense temporal transforms (slowed down). RTR fails to detect this region while our model is able to capture it well.

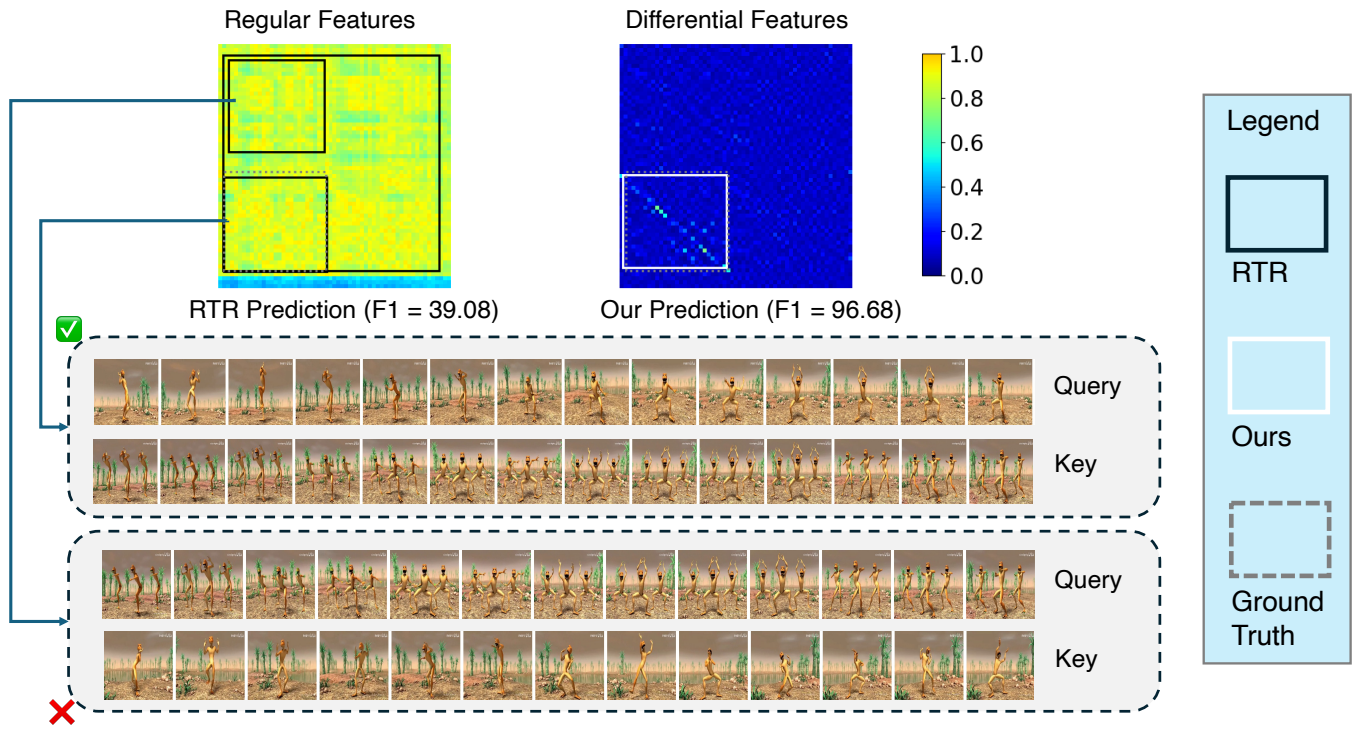


Figure 9: An example of video copy localization predictions made by the RTR model (left) and our model (right). We show the RTR prediction using black bounding boxes and our predictions using white bounding boxes. The ground-truth label is shown using gray dotted bounding boxes. The cosine similarity map is visualized on the left and the differential map visualized on the right. Below, we show the video frames for the section of video which is correctly predicted by both models (top), and wrongly predicted by RTR (bottom). The video depicts a character dancing. We see that the wrongly predicted section also exhibits very high similarity on the cosine-similarity map. However, the differential similarity map picks up a clearer signal from the actual copied regions, which presents a better measure of similarity based on the correspondence of the frames within each video. Although the overall scene is highly similar, the differential features and our DiF-SiM help pick up the change in motion between frames for more accurate copy localization.