

Supplementary Materials: Not All Pairs are Equal: Hierarchical Learning for Average-Precision-Oriented Video Retrieval

Anonymous Authors

CONTENTS

A Additional Illustration of Method	1
A.1 Derivation of AP Risk	1
A.2 Proofs of QuadLinear-AP's properties	1
A.2.1 Differentiability	1
A.2.2 Smoothness	2
A.2.3 Convexity	2
A.2.4 Non-strictly Monotonically Increasing	2
A.2.5 Upper Bound of Heaviside Function	2
A.3 Description of the Basic Loss	2
B Detailed Description of Experiments	3
B.1 Datasets	3
B.2 Evaluation Metrics	3
B.3 Implementation Details	3
B.4 Additional Ablation Study	4
C Additional Visualization	6
References	6

A ADDITIONAL ILLUSTRATION OF METHOD

A.1 Derivation of AP Risk

In a batch of videos $B = \{V_i \in \mathcal{X}\}_{i=1}^N$, recall that for a query video V_k , the similarity scores of the relevant/irrelevant videos are denoted as S^{k+}/S^{k-} . For simplicity, let $d_{ji}^k = s_{kj} - s_{ki}$. As mentioned in section 3.1, our goal is to maximize the AP score. This is achieved by minimizing the AP risk, which is derived as follows:

$$\begin{aligned}
 AP_k^{\downarrow}(f) &= 1 - AP_k(f) \\
 &= 1 - \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \frac{\mathcal{R}(s_{ki}, S^{k+})}{\mathcal{R}(s_{ki}, S^{k+} \cup S^{k-})} \\
 &= 1 - \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \frac{1 + \sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)}{1 + \sum_{s_{kj} \in S^{k+} \cup S^{k-}} \mathcal{H}(d_{ji}^k)} \\
 &= \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \frac{\sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)}{1 + \sum_{s_{kj} \in S^{k+} \cup S^{k-}} \mathcal{H}(d_{ji}^k)} \\
 &= \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \frac{\sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)}{1 + \sum_{s_{kj} \in S^{k+}} \mathcal{H}(d_{ji}^k) + \sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)} \\
 &= \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \frac{[\sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)] / [1 + \sum_{s_{kj} \in S^{k+}} \mathcal{H}(d_{ji}^k)]}{1 + [\sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)] / [1 + \sum_{s_{kj} \in S^{k+}} \mathcal{H}(d_{ji}^k)]} \\
 &= \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} h\left(\frac{\sum_{s_{kj} \in S^{k-}} \mathcal{H}(d_{ji}^k)}{1 + \sum_{s_{kj} \in S^{k+}} \mathcal{H}(d_{ji}^k)}\right),
 \end{aligned}$$

where $\mathcal{R}(s, S) = 1 + \sum_{s' \in S} \mathcal{H}(s' - s)$ is the descending ranking of s in S , $\mathcal{H}(\cdot)$ is the Heaviside function, $h(x) = \frac{x}{1+x}$ is a monotonically increasing function.

We substitute the Heaviside function in the numerator with $\mathcal{R}^-(d_{ji}^k; \delta)$ in eq.(10) and introduce an additional parameter ρ , which forms the following surrogate AP risk:

$$\widehat{AP}_k^{\downarrow}(f) = \frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} h\left(\frac{\sum_{s_{kj} \in S^{k-}} \mathcal{R}^-(d_{ji}^k; \delta)}{1 + \rho \sum_{s_{kj} \in S^{k+}} \mathcal{H}(d_{ji}^k)}\right).$$

A.2 Proofs of QuadLinear-AP's properties

In this subsection, we provide proofs for several properties of QuadLinear-AP as outlined in section 3.3 of the main paper. Specifically, we focus on the proofs of $\mathcal{R}^-(x; \delta)$ since it determines these properties of QuadLinear-AP.

A.2.1 Differentiability. Note that it is unnecessary to replace $\mathcal{H}(\cdot)$ for the positive-positive pair since it only plays a role of weight for precisely measuring each term in eq. (8). Therefore, we only need to ensure the $\mathcal{R}^-(x; \delta)$ is differentiable, which is proved as follows.

First, the $\mathcal{R}^-(x; \delta)$ can be reformatted as:

$$\mathcal{R}^-(x; \delta) = \begin{cases} \frac{2}{\delta}x + 1, & \text{if } x \geq 0. \\ \frac{1}{\delta^2}x^2 + \frac{2}{\delta}x + 1, & \text{if } -\delta \leq x < 0. \\ 0, & \text{if } x < -\delta. \end{cases}$$

Clearly, $\mathcal{R}^-(x; \delta)$ is differentiable on its three segments. Now, we only need to verify that it is differentiable at the points where $x = -\delta$ and $x = 0$.

When $x = -\delta$, we have:

$$\begin{aligned}
 \frac{d\mathcal{R}^-(x^-; \delta)}{dx^-} &= \lim_{x^- \rightarrow -\delta} \frac{\mathcal{R}^-(x^-; \delta) - \mathcal{R}^-(-\delta; \delta)}{x^- - (-\delta)} = 0, \\
 \frac{d\mathcal{R}^-(x^+; \delta)}{dx^+} &= \lim_{x^+ \rightarrow -\delta} \frac{\mathcal{R}^-(x^+; \delta) - \mathcal{R}^-(-\delta; \delta)}{x^+ - (-\delta)} = 0, \\
 \frac{d\mathcal{R}^-(x^-; \delta)}{dx^-} &= \frac{d\mathcal{R}^-(x^+; \delta)}{dx^+} = \frac{d\mathcal{R}^-(x; \delta)}{dx} \Big|_{x=-\delta} = 0.
 \end{aligned}$$

When $x = 0$, we have:

$$\begin{aligned}
 \frac{d\mathcal{R}^-(x^-; \delta)}{dx^-} &= \lim_{x^- \rightarrow 0} \frac{\mathcal{R}^-(x^-; \delta) - \mathcal{R}^-(0; \delta)}{x^- - 0} = \frac{2}{\delta}, \\
 \frac{d\mathcal{R}^-(x^+; \delta)}{dx^+} &= \lim_{x^+ \rightarrow 0} \frac{\mathcal{R}^-(x^+; \delta) - \mathcal{R}^-(0; \delta)}{x^+ - 0} = \frac{2}{\delta}, \\
 \frac{d\mathcal{R}^-(x^-; \delta)}{dx^-} &= \frac{d\mathcal{R}^-(x^+; \delta)}{dx^+} = \frac{d\mathcal{R}^-(x; \delta)}{dx} \Big|_{x=0} = \frac{2}{\delta}.
 \end{aligned}$$

Therefore, it is proven that $\mathcal{R}^-(x; \delta)$ is differentiable at each point, allowing backpropagation to be performed effectively during the optimization process to update model parameters. The derivative function of $\mathcal{R}^-(x; \delta)$ can be formulated as follows:

$$\frac{d\mathcal{R}^-(x; \delta)}{dx} = \begin{cases} \frac{2}{\delta}, & \text{if } x \geq 0. \\ \frac{2}{\delta^2}x + \frac{2}{\delta}, & \text{if } -\delta \leq x < 0. \\ 0, & \text{if } x < -\delta. \end{cases}$$

A.2.2 Smoothness. To prove the smoothness of $\mathcal{R}^-(x; \delta)$ is equivalent to proving that the derivation function of $\mathcal{R}^-(x; \delta)$ is continuous. This continuity is essential for ensuring stable gradient changes for efficient optimization and smooth convergence of the model. For the sake of presentation, let $\mathcal{D}^-(x; \delta) = \frac{d\mathcal{R}^-(x; \delta)}{dx}$.

Clearly, $\mathcal{D}^-(x; \delta)$ is continuous on its three segments, thus we only need to verify that it is continuous at the points where $x = -\delta$ and $x = 0$, which is presented as follows:

$$\lim_{x^- \rightarrow -\delta} \mathcal{D}^-(x; \delta) = \lim_{x^+ \rightarrow -\delta} \mathcal{D}^-(x; \delta) = \mathcal{D}^-(x; \delta) = 0,$$

$$\lim_{x^- \rightarrow 0} \mathcal{D}^-(x; \delta) = \lim_{x^+ \rightarrow 0} \mathcal{D}^-(x; \delta) = \mathcal{D}^-(0; \delta) = \frac{2}{\delta}.$$

Therefore, it is proven that $\mathcal{R}^-(x; \delta)$ is smooth, and the derivative function of $\mathcal{R}^-(x; \delta)$ is continuous at each point.

A.2.3 Convexity. First, it is obvious that $\mathcal{R}^-(x; \delta)$ is convex on its three segments, thus we only need to verify three situations by proving $t\mathcal{R}^-(x_1; \delta) + (1-t)\mathcal{R}^-(x_2; \delta) - \mathcal{R}^-(tx_1 + (1-t)x_2; \delta) \geq 0$ for the given $0 \leq t \leq 1$ and $x_1 < x_2$.

1) When $-\delta \leq x_1 < 0 \leq x_2$, we have:

$$t\mathcal{R}^-(x_1; \delta) + (1-t)\mathcal{R}^-(x_2; \delta) = t \left(\frac{1}{\delta^2}x_1^2 + \frac{2}{\delta}x_1 + 1 \right) + (1-t) \left(\frac{2}{\delta}x_2 + 1 \right).$$

If $tx_1 + (1-t)x_2 \geq 0$ then:

$$\begin{aligned} \mathcal{R}^-(tx_1 + (1-t)x_2; \delta) &= \frac{2}{\delta} [tx_1 + (1-t)x_2] + 1. \\ t\mathcal{R}^-(x_1; \delta) + (1-t)\mathcal{R}^-(x_2; \delta) - \mathcal{R}^-(tx_1 + (1-t)x_2; \delta) \\ &= \frac{t}{\delta^2}x_1^2 \\ &> 0. \end{aligned}$$

If $-\delta \leq tx_1 + (1-t)x_2 < 0$ then:

$$\begin{aligned} \mathcal{R}^-(tx_1 + (1-t)x_2; \delta) &= \left\{ \frac{1}{\delta} [tx_1 + (1-t)x_2] + 1 \right\}^2. \\ t\mathcal{R}^-(x_1; \delta) + (1-t)\mathcal{R}^-(x_2; \delta) - \mathcal{R}^-(tx_1 + (1-t)x_2; \delta) \\ &= \frac{t}{\delta^2}x_1^2 - \frac{1}{\delta^2} [tx_1 + (1-t)x_2]^2 \\ &= \left[\frac{\sqrt{t}}{\delta}x_1 - \frac{t}{\delta}x_1 - \frac{1-t}{\delta}x_2 \right] \left[\frac{\sqrt{t}}{\delta}x_1 + \frac{t}{\delta}x_1 + \frac{1-t}{\delta}x_2 \right] \\ &> \left[\frac{\sqrt{t}}{\delta}x_1 - \frac{1}{\delta}x_2 \right] \left[\frac{\sqrt{t}}{\delta}x_1 + \frac{1}{\delta}x_1 \right] \\ &> 0. \end{aligned}$$

For the other two situations, i.e., $x_1 < -\delta < 0 \leq x_2$ and $x_1 < -\delta \leq x_2 < 0$, the proof process is similar to the situation discussed above, and is therefore omitted for brevity.

In summary, $\mathcal{R}^-(x; \delta)$ is convex at each point, which facilitates finding the optimal solution while maintaining good convergence speed and stability.

A.2.4 Non-strictly Monotonically Increasing. First, it is obvious that $\mathcal{R}^-(x; \delta)$ is non-strictly monotonically increasing on its three segments, thus we only need to verify the following three situations:

1) When $-\delta \leq x < 0$, for given $\varepsilon > 0$, if $x + \varepsilon \geq 0$ we have:

$$\begin{aligned} \mathcal{R}^-(x + \varepsilon; \delta) - \mathcal{R}^-(x; \delta) &= \frac{2}{\delta}(x + \varepsilon) + 1 - \left(\frac{1}{\delta^2}x^2 + \frac{2}{\delta}x + 1 \right) \\ &= \frac{1}{\delta} \left(2\varepsilon - \frac{1}{\delta}x^2 \right) \\ &\geq \frac{1}{\delta} \left(-2x + \frac{1}{x} \cdot x^2 \right) \\ &> 0. \end{aligned}$$

2) When $x < -\delta$, for given $\varepsilon > 0$, if $-\delta \leq x + \varepsilon < 0$ we have:

$$\begin{aligned} \mathcal{R}^-(x + \varepsilon; \delta) - \mathcal{R}^-(x; \delta) &= \frac{1}{\delta^2}(x + \varepsilon)^2 + \frac{2}{\delta}(x + \varepsilon) + 1 \\ &= \left[\frac{1}{\delta}(x + \varepsilon) \right]^2 \\ &> 0. \end{aligned}$$

3) When $x < -\delta$, for given $\varepsilon > 0$, if $x + \varepsilon \geq 0$ we have:

$$\mathcal{R}^-(x + \varepsilon; \delta) - \mathcal{R}^-(x; \delta) = \frac{2}{\delta}(x + \varepsilon) + 1 > 0.$$

In summary, $\mathcal{R}^-(x; \delta)$ is non-strictly monotonically increasing, which can also be supported by the fig. 4c in the main paper.

A.2.5 Upper Bound of Heaviside Function. Here we prove $\mathcal{R}^-(x; \delta)$ is the upper bound of $\mathcal{H}(x)$, which is equivalent to prove the $\mathcal{R}^-(x; \delta) - \mathcal{H}(x) \geq 0$. Let $\mathcal{P}^-(x; \delta) = \mathcal{R}^-(x; \delta) - \mathcal{H}(x)$, we have:

$$\mathcal{P}^-(x; \delta) = \begin{cases} \frac{2}{\delta}x, & \text{if } x \geq 0. \\ \frac{1}{\delta^2}x^2 + \frac{2}{\delta}x + 1, & \text{if } -\delta \leq x < 0. \\ 0, & \text{if } x < -\delta. \end{cases}$$

Obviously, $\mathcal{P}^-(x; \delta) \geq 0$, which illustrates the $\mathcal{R}^-(x; \delta)$ is the upper bound of $\mathcal{H}(x)$.

A.3 Description of the Basic Loss

As outlined in section 3.4, following previous methods on ranking optimization [3, 21], we combine the AP losses with a basic loss \mathcal{L}_{base} , which comprises the InfoNCE loss [16] and an SSHN loss [11].

The InfoNCE loss is widely used in self-supervised contrastive learning tasks due to its effectiveness and adaptability. For a query video V_k , the InfoNCE loss is calculated by:

$$\mathcal{L}_{NCE}^k = -\frac{1}{|S^{k+}|} \sum_{s_{ki} \in S^{k+}} \log \frac{\exp(s_{ki}/\tau)}{\exp(s_{ki}/\tau) + \sum_{s_{kj} \in S^{k-}} \exp(s_{kj}/\tau)}.$$

Using the InfoNCE enables the model to support representation learning by distinguishing between positive and negative instances, thus promoting collaborative optimization between ranking and representation learning.

The SSHN loss promotes self-similarity towards 1 by compensating for the CNN block ψ , which tends to make s_{kk} less than 1.

Additionally, it performs hard negative mining by reducing the similarity of the most challenging negative instances, thus enhancing the distinction between similarities. The SSHN loss can be formulated as follows:

$$\mathcal{L}_{SSH}^k = -\log(s_{kk}) - \log\left(\max_{s_{ki} \in S^{k-}} (1 - s_{ki})\right).$$

Finally, we integrate these two losses as the following basic loss function, where λ_s is hyperparameters to adjust the weights of the two losses.

$$\mathcal{L}_{base} = \frac{1}{N} \sum_{k=1}^N (\mathcal{L}_{NCE}^k + \lambda_s \mathcal{L}_{SSH}^k). \quad (1)$$

B DETAILED DESCRIPTION OF EXPERIMENTS

B.1 Datasets

The detailed description of the datasets used in our experiments is as follows:

- **VCDB** [8] is designed for the task of partial video copy detection. It contains a labeled core dataset denoted as VCDB (C) and a large-scale unlabeled dataset with 100,000 distractor videos denoted as VCDB (D). In our experiments, we only use the VCDB (D) for self-supervised training of the model.
- **EVVE** [20] is used as a benchmark video dataset for the task of event-based video retrieval. It includes 620 query videos and 2,373 database videos manually annotated into 13 event categories. Due to the absence of some videos, only 504 query videos and 1,906 database videos can be obtained.
- **SVD** [7] is designed for the task of near-duplicate video retrieval, containing 1,206 queries and 526,787 unlabelled videos in total. The dataset is organized into a training set and a test set. For evaluation in the experiments, we exclusively employ the test set, which includes 206 queries with 6,355 labeled video pairs and 526,787 unlabelled videos as distractors.
- **FIVR-200K** [9] is specifically designed for fine-grained incident video retrieval, comprising 100 queries and 225,960 database videos. It contains three distinct video retrieval subtasks: Duplicate Scene Video Retrieval (DSVR), Complementary Scene Video Retrieval (CSVR), and Incident Scene Video Retrieval (ISVR). Additionally, it also contains three distinct video detection subtasks: Duplicate Scene Video Detection (DSVD), Complementary Scene Video Detection (CSVD), and Incident Scene Video Detection (ISVD).
- **FIVR-5K** [11], a subset of FIVR-200K, which includes 50 queries and 5,000 database videos, containing the same subtasks as FIVR-200K. This dataset is also utilized in our experiments to facilitate swift comparative analysis.

Generally, we use the origin videos from VCDB (D) to train our model and conduct evaluation on EVVE, SVD, FIVR-200K as well as FIVR-5K. Following the previous works [11], we use the extracted features of the evaluation datasets in our experiments. A summary of the descriptions for these datasets is presented in table 1.

B.2 Evaluation Metrics

Mean Average Precision. Mean Average Precision (**mAP**), also known as macro Average Precision [18], serves as the primary metric to evaluate the overall performance of retrieval tasks. Specifically, AP computes the average ranking of positive instances in the retrieval set for a particular query, while mAP calculates the mean of these AP values across all queries. The definition of mAP is given in eq. (2), where n_j denotes the number of positive instances for a particular query, r_i represents the ranking of the i -th retrieved positive instance in the retrieval set, and $|Q|$ is the number of query instances.

$$mAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{n_j} \sum_{i=1}^{n_j} \frac{i}{r_i} \quad (2)$$

Micro Average Precision. Micro Average Precision (**μ AP**) is a metric employed in prior research [11, 13, 19] to evaluate the performance of detection tasks. In contrast to mAP, μ AP considers the joint distribution of similarities across all queries by calculating the AP across all queries simultaneously, which reflects the model's capability to consistently apply a uniform threshold across various queries to detect relevant instances. μ AP is computed as outlined in eq. (3), where $|R|$ is the number of all reference instances, $p(i)$ represents the precision of i -th instance and $\Delta r(i)$ denotes the difference of recall between i -th and its adjacent instance in the sorted list according to similarity scores.

$$\mu AP = \sum_{i=1}^{|R|} p(i) \Delta r(i) \quad (3)$$

B.3 Implementation Details

In this subsection, we provide additional descriptions of implementation details including the data processing, experiment configuration, and hyperparameter settings.

Data processing. We adopt a self-supervised learning approach as introduced in [11], where videos in a batch are subjected to weak and strong augmentations to simulate common video copy transformations in actual situations. The weak augmentation function set A_w includes traditional geometric transformations such as random cropping and horizontal flipping, applied to the frames of the entire video. The strong augmentation function set A_s , on the other hand, involves more complex transformations: **1) Global transformations** apply different geometric and optical image transformations on all frames consistently by RandAugment [2]; **2) Frame transformations** encompass overlaying emojis and text on randomly selected frames and applying blur to frames [19]; **3) Temporal transformations**, including fast forward, slow motion, reverse play, frame pause, and sub-clip shuffle/dropout [10, 11], are utilized to create intense temporal manipulations; **4) Video mix-up transformation**, down-scales a video and embeds it within another video [11].

Experiment configuration. For the training video data, following the previous work [10, 12, 22], we first extract one frame per second for each video. Subsequently, we resize the frames to 256 pixels and crop them to 224 pixels, then randomly select 28 consecutive frames to constitute a video clip. For the backbone feature extractor $g(\cdot)$, following previous literature [10–12], we adopt

Table 1: Summary of the descriptions for the VCDB, EVVE, SVD, FIVR-200K, and FIVR-5K datasets.

Dataset	Video Task	# of Query Videos	# of Database Videos
VCDB	Partial Video Copy Detection	528	100,000
EVVE	Event-based Video Retrieval	620	2,373
SVD	Near-duplicate Video Retrieval	206	526,787
FIVR-200K DSVR / DSVD	Duplicate Scene Video Retrieval / Detection	200	225,960
FIVR-200K CSVr / CSVD	Complementary Scene Video Retrieval / Detection	200	225,960
FIVR-200K ISVR / ISVD	Incident Scene Video Retrieval / Detection	200	225,960
FIVR-5K DSVR / DSVD	Duplicate Scene Video Retrieval / Detection	50	5,000
FIVR-5K CSVr / CSVD	Complementary Scene Video Retrieval / Detection	50	5,000
FIVR-5K ISVR / ISVD	Incident Scene Video Retrieval / Detection	50	5,000

ResNet50 [6] pretrained on ImageNet [4]. The backbone feature extractor $g(\cdot)$ performs the mapping $g : \mathbb{R}^{T \times H \times W \times C} \rightarrow \mathbb{R}^{T \times R \times D}$, where $T = 28, H = 224, W = 224, C = 3, R = 9, D = 512$. For the feature extractor $g'(\cdot)$ in the pseudo label generator, we utilize DINO [1] pretrained ViT-small [5] with a patch size of 16. The feature extractor $g'(\cdot)$ performs the mapping $g' : \mathbb{R}^{T \times H \times W \times C} \rightarrow \mathbb{R}^{T \times D'}$, where $T = 28, H = 224, W = 224, C = 3, D' = 384$.

Hyperparameter settings. Our model is trained for 30,000 iterations with a batch size of 64. We use AdamW [15] with the Cosine Annealing scheduler for parameters optimization. The learning rate is set to 4×10^{-5} with a warm-up period [14] of 1,000 iterations, and weight decay is set to 1×10^{-2} . For the hyperparameters concerning QuadLinear-AP, we choose $\delta_v = 0.05, \rho_v = 0.10$ for \mathcal{L}_{QLAP}^V , and $\delta_f = 0.05, \rho_f = 5.00$ for \mathcal{L}_{QLAP}^F . The weights of \mathcal{L}_{QLAP}^V and \mathcal{L}_{QLAP}^F are selected as $\lambda_v = 4$ and $\lambda_f = 6$, respectively. The top and bottom rates for dividing positive and negative frame instances in the pseudo label generator are set to $r_t = 0.35$ and $r_b = 0.35$. The top-k rates of TopK-Chamfer Similarity within spacial and temporal correlation aggregation are set to $k_s = 0.10$ and $k_t = 0.03$, respectively.

Generally, the settings and hyperparameters for our HAP-VR framework within the training process are summarized in table 2. All experiments in this work are conducted with Pytorch [17] library on a Linux machine equipped with an Intel Gold 6230R CPU and two NVIDIA 3090 GPUs.

B.4 Additional Ablation Study

In this section, we explore the impact of hyperparameters in our framework on performance. Except for the specific hyperparameters being investigated, we maintain consistency in all other experimental settings to ensure a fair comparison.

Impact of δ_v and δ_f . The results of our model trained with different δ_v and δ_f are presented in table 3 and table 4, respectively. The performance decreases for both hyperparameters when set above or below 0.05. This highlights the importance of selecting the appropriate δ_v and δ_f values to effectively balance the margin for correctly ranked positive-negative pairs and the penalty for incorrectly ranked positive-negative pairs.

Impact of ρ_v and ρ_f . The results of our model trained with different ρ_v and ρ_f are presented in table 5 and table 6, respectively.

Table 2: The settings and hyperparameters for our HAP-VR framework within the training process.

Hyperparameter	Notation	Value
Training process		
Iterations	/	30,000
Warm-up iterations	/	1,000
Batch size	/	64
Learning rate	/	4×10^{-5}
Optimizer	/	AdamW
Learning rate scheduler	/	Cosine
Weight decay	/	1×10^{-2}
Backbone feature extractor		
# of frames in a clip	T	28
Frame size	H, W	224
# of ResNet50 feature patch	R	9
# of ResNet50 feature dim.	D	512
Pseudo label generator		
# of frames in a clip	T	28
Frame size	H, W	224
# of ViT-small feature dim.	D'	384
ViT-small patch size	/	16
Top frame rate	r_t	0.35
Bottom frame rate	r_b	0.35
QuadLinear-AP		
Video-level Pos-neg margin	δ_v	0.05
Video-level Pos-pos weight	ρ_v	0.10
Video-level AP loss weight	λ_v	4.00
Frame-level Pos-neg margin	δ_f	0.05
Frame-level Pos-pos weight	ρ_f	5.00
Frame-level AP loss weight	λ_f	6.00
TopK-Chamfer Similarity		
Spacial top-k rate	k_s	0.10
Temporal top-k rate	k_t	0.03

For \mathcal{L}_{QLAP}^V , setting ρ_v as a small value such as 0.2 provides optimal benefits as it more effectively adjusts the weight of positive-positive pairs and thus achieves a trade-off with positive-negative pairs. For \mathcal{L}_{QLAP}^F , which deals with more ambiguous inter-frame correlations, tuning ρ_f within the range of 0.2 to 5 allows the model to better adapt to the varying distributions of positive and negative instances across different subtasks.

Impact of λ_v and λ_f . In table 7, we report the results of our model trained with various values of λ_f while keeping λ_v fixed at 4 to simplify the comparative analysis. It can be observed that increasing λ_f beyond λ_v leads to an obvious performance gain. This is expected as more challenging frame-level similarities require greater weight for effective optimization. Furthermore, finding a balance between λ_f and λ_v with the weight of \mathcal{L}_{base} can jointly promote ranking and representation learning, thereby enhancing the overall performance of the model.

Impact of r_t and r_b . In table 8, we report the results of our model trained with various combinations of r_t and r_b . When r_t and r_b are equal, setting higher values leads to similar frames being forcibly divided as positive and negative instances, thereby decreasing the model's discriminating ability. Conversely, setting lower values may cause the model to focus only on easier instances, thus resulting in insufficient learning and optimization. When r_t and r_b are different, the performance tends to decrease due to the uneven distribution of positive and negative instances increasing the complexity of similarity learning.

Table 3: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for δ_v within \mathcal{L}_{QLAP}^V . The first and second best results are marked with bold and underline.

δ_v	Retrieval			Detection		
	DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
0.01	87.32	86.66	80.27	73.53	72.15	63.58
0.05	88.86	87.79	80.34	78.15	76.29	65.88
0.10	<u>88.52</u>	<u>87.38</u>	<u>80.37</u>	<u>76.94</u>	<u>75.40</u>	<u>65.60</u>
0.15	87.34	86.62	80.42	73.27	72.37	64.11

Table 4: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for δ_f within \mathcal{L}_{QLAP}^F . The first and second best results are marked with bold and underline.

δ_f	Retrieval			Detection		
	DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
0.01	<u>90.30</u>	89.19	81.52	81.17	78.37	69.98
0.05	90.37	<u>89.11</u>	80.94	83.62	80.47	69.30
0.10	89.85	88.52	79.93	<u>82.29</u>	<u>79.57</u>	67.15
0.15	89.55	88.05	79.37	<u>82.29</u>	79.40	65.78

Table 5: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for ρ_v within \mathcal{L}_{QLAP}^V . The first and second best results are marked with bold and underline.

ρ_v	Retrieval			Detection		
	DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
0.02	89.52	88.45	80.84	78.34	76.39	65.98
0.2	90.28	89.10	81.09	81.39	78.64	<u>68.79</u>
1.0	<u>89.89</u>	<u>88.81</u>	<u>81.03</u>	<u>80.41</u>	78.64	69.41
5.0	89.51	88.31	80.77	80.00	<u>77.59</u>	67.20
50	89.37	88.16	80.89	79.92	77.19	66.91

Table 6: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for ρ_f within \mathcal{L}_{QLAP}^F . The first and second best results are marked with bold and underline.

ρ_f	Retrieval			Detection		
	DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
0.02	89.71	88.01	79.44	82.73	78.98	65.80
0.2	<u>90.21</u>	89.01	80.92	83.81	80.59	69.13
1.0	90.37	89.11	80.94	<u>83.62</u>	<u>80.47</u>	69.30
5.0	90.17	<u>89.07</u>	81.29	82.93	80.18	70.99
50	88.77	87.74	<u>81.18</u>	79.76	78.46	<u>70.73</u>

Table 7: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for the weight of \mathcal{L}_{QLAP}^F , i.e., λ_f . The first and second best results are marked with bold and underline.

λ_f	Retrieval			Detection		
	DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
2	89.73	88.50	80.42	80.45	77.89	66.32
4	<u>90.21</u>	<u>89.08</u>	<u>81.33</u>	<u>83.15</u>	<u>80.69</u>	<u>70.61</u>
6	90.26	89.18	81.47	83.05	80.11	70.32
8	90.07	88.91	81.30	84.20	80.97	70.63

Table 8: Results on FIVR-5K in video retrieval and detection tasks with mAP (%) and μ AP (%) for r_t and r_b . The first and second best results are marked with bold and underline.

r_t	r_b	Retrieval			Detection		
		DSVR	CSVR	ISVR	DSVD	CSVD	ISVD
0.30	0.30	90.17	88.92	80.90	83.08	80.10	69.02
0.35	0.35	90.21	89.08	81.33	83.15	80.69	<u>70.61</u>
0.40	0.40	<u>90.17</u>	<u>89.07</u>	<u>81.29</u>	82.93	<u>80.18</u>	70.99
0.45	0.45	89.63	88.37	80.89	82.28	79.38	70.18
0.30	0.40	89.73	88.62	81.18	82.67	79.68	69.18
0.40	0.30	89.82	88.58	80.62	82.28	79.33	69.18

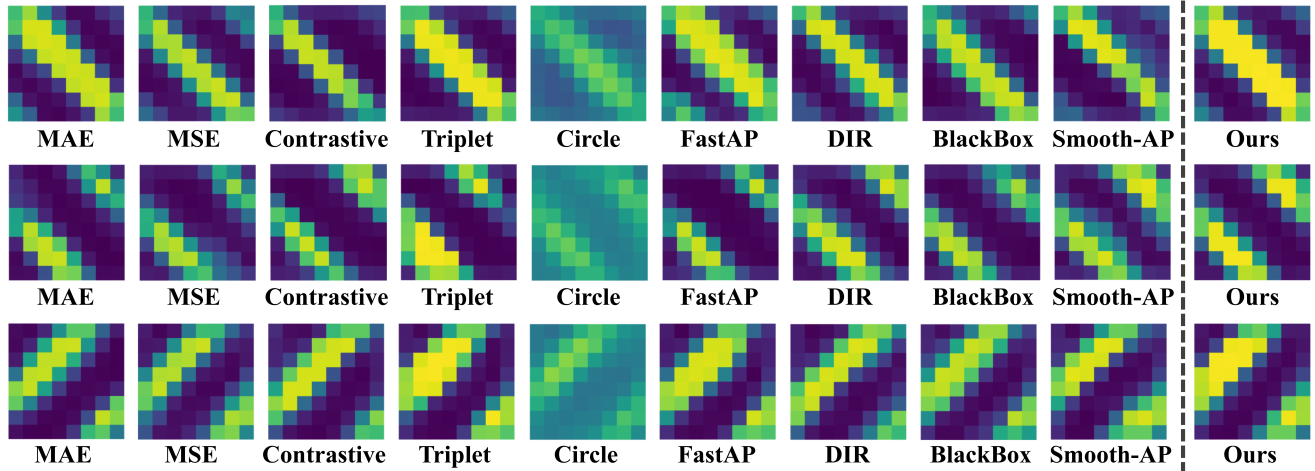


Figure 1: Heatmaps of frame-level similarity matrices generated by various losses. In contrast, our QuadLinear-AP distinguishes between relevant and irrelevant instances more clearly. A brighter color indicates a higher similarity score.

C ADDITIONAL VISUALIZATION

In this section, we provide additional examples to compare frame-level similarity matrices under different losses through visualization for intuitive analysis. The experiment settings remain consistent with those described in section 4.3 of the main paper. By analyzing these heatmaps, we can make the following observations: 1) Circle loss struggles to distinguish instances clearly even after parameter adjustments, likely due to its sensitivity of data distribution making it perform poorly in challenging video data with an imbalanced distribution. 2) Triplet loss is prone to become overconfident, which may lead to more irrelevant instances being predicted as relevant, thus increasing the risk of overfitting. 3) While other loss functions can discriminate between instances, there is still room for improvement in their performance. 4) Compared to other competitors, our proposed QuadLinear-AP provides a clearer distinction between relevant and irrelevant instances, making it effective for video retrieval tasks.

REFERENCES

- [1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*. 9650–9660.
- [2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*. 702–703.
- [3] Siran Dai, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2024. DRAUC: An Instance-wise Distributionally Robust AUC Optimization Framework. *Advances in Neural Information Processing Systems* 36 (2024).
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li-Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Ieee, 248–255.
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 770–778.
- [7] Qing-Yuan Jiang, Yi He, Gen Li, Jian Lin, Lei Li, and Wu-Jun Li. 2019. SVD: A large-scale short video dataset for near-duplicate video retrieval. In *International Conference on Computer Vision*. 5281–5289.
- [8] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. 2014. VCDDB: a large-scale database for partial copy detection in videos. In *European Conference on Computer Vision*. Springer, 357–371.
- [9] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. FIVR: Fine-grained incident video retrieval. *IEEE Transactions on Multimedia* 21, 10 (2019), 2638–2652.
- [10] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. Visil: Fine-grained spatio-temporal video similarity learning. In *International Conference on Computer Vision*. 6351–6360.
- [11] Giorgos Kordopatis-Zilos, Giorgos Tolias, Christos Tzelepis, Ioannis Kompatsiaris, Ioannis Patras, and Symeon Papadopoulos. 2023. Self-Supervised Video Similarity Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4755–4765.
- [12] Giorgos Kordopatis-Zilos, Christos Tzelepis, 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.
- [13] Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Olivier Buisson, Valerie Gouet-Brunet, Nozha Boujemaa, and Fred Stentiford. 2007. Video copy detection: a comparative study. In *Proceedings of the 6th ACM international conference on Image and video retrieval*. 371–378.
- [14] Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*.
- [15] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- [16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
- [17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* 32 (2019).
- [18] Florent Perronnin, Yan Liu, and Jean-Michel Renders. 2009. A family of contextual measures of similarity between distributions with application to image retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2358–2365.
- [19] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. 2022. A self-supervised descriptor for image copy detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14532–14542.
- [20] Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. 2013. Event retrieval in large video collections with circulant temporal encoding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2459–2466.
- [21] Huiyang Shao, Qianqian Xu, Zhiyong Yang, Peisong Wen, Gao Peifeng, and Qingming Huang. 2024. Weighted roc curve in cost space: Extending auc to cost-sensitive learning. *Advances in Neural Information Processing Systems* 36 (2024).
- [22] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. 2021. Temporal context aggregation for video retrieval with contrastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 3268–3278.