# Learning Partially-Decorrelated Common Spaces for Ad-hoc Video Search

Fan Hu
Renmin University of China
Beijing, China
hufan_hf@ruc.edu.cn

Zijie Xin
Renmin University of China
Beijing, China
xinzijie@ruc.edu.cn

Xirong Li*
Renmin University of China
Beijing, China
xirong@ruc.edu.cn

## Abstract

Ad-hoc Video Search (AVS) involves using a textual query to search for *multiple* relevant videos in a large collection of unlabeled short videos. The main challenge of AVS is the visual diversity of relevant videos. A simple query such as "Find shots of a man and a woman dancing together indoors" can span a multitude of environments, from brightly lit halls and shadowy bars to dance scenes in black-and-white animations. It is therefore essential to retrieve relevant videos as comprehensively as possible. Current solutions for the AVS task primarily fuse multiple features into one or more common spaces, yet overlook the need for diverse spaces. To fully exploit the expressive capability of individual features, we propose LPD, short for **L**earning **P**artially **D**ecorrelated common spaces. LPD incorporates two key innovations: feature-specific common space construction and the de-correlation loss. Specifically, LPD learns a separate common space for each video and text feature, and employs de-correlation loss to diversify the ordering of negative samples across different spaces. To enhance the consistency of multi-space convergence, we designed an entropy-based fair multi-space triplet ranking loss. Extensive experiments on the TRECVID AVS benchmarks (2016-2023) justify the effectiveness of LPD. Moreover, diversity visualizations of LPD's spaces highlight its ability to enhance result diversity.

## CCS Concepts

• **Information systems** → **Video search**; **Information retrieval diversity**.

## Keywords

Ad-hoc video search, multi-feature fusion, diverse common space

*Corresponding author.

## 1 Introduction

Ad-hoc video search (AVS), or zero-example video retrieval, involves identifying multiple relevant videos through text-based queries without visual references. Since 2016, the annual TRECVID (TV) evaluation has been a benchmark for measuring progress on the AVS task [1]. Each team is asked to develop a video retrieval system that retrieves the top 1,000 items for each test query from a large collection of unlabeled short videos. The organizers then annotate the items submitted by all teams and announce the scores and rankings. AVS is different from text-to-video matching (retrieval) task, which seeks to identify the *most* semantically relevant video clip from a candidate pool referring to the textual query [2–5]. AVS aims to retrieve all videos related to the textual query, which may encompass a variety of scenarios [6–12]. The visualization examples between these two tasks are illustrated in Fig. 1.

The AVS task is particularly challenging for two key reasons. Firstly, it requires searching from an extremely large collection of unlabeled short videos. The size of the test collection for AVS TV 2016–2018 (IACC.3 dataset [1]) is 336k, which increases to one million for TV 2019–2021 (V3C1 dataset [14]) and 1.5 million for TV 2022–2023 (V3C2 dataset [11]). The second challenge is that AVS task not only aims to accurately locate videos that match the query, but also comprehensively retrieve relevant videos across all possible scenes. The complexity is heightened by the potential variability in the relevant video content, including scenes set in broad daylight, during the night, or even within animated environments. As shown in Fig. 1, a simple query like "*Find shots of a man and a woman dancing together indoors*" could span a multitude of environments, from brightly lit halls and shadowy bars to dance scenes within black and white animations, which highlight the assessment challenges outlined in annual TRECVID reports [1, 6–12].

Currently, the end-to-end task-specific models based on CLIP [15] are popular in text-to-video matching task [3, 4, 16–18]. However, given the considerable challenges posed by the large size of video pools and the diversity of video scenes, the winning models of AVS task are primarily based on deep learning models that extract diverse (off-the-shell) textual or video features, followed by cross-modal training and matching [19–29]. We categorize them into two groups based on the number of common spaces:

The first category is learning one common space with multiple feature fusion, such as DE [19, 22, 30–32] and W2VV++ [19, 31, 33], fusing multiple features/models to create a stronger composite feature. However, learning within a singular common space could present challenges in achieving diversity and comprehensive recall. The second category is centered around learning multiple common spaces with different features, such as creating a concept space and an embedding space [21, 22, 34], multi-level video-text

**(a) Text-to-video matching. Data: MSR-VTT [13]**



**(b) Ad-hoc video search. Data: TRECVID [8]**
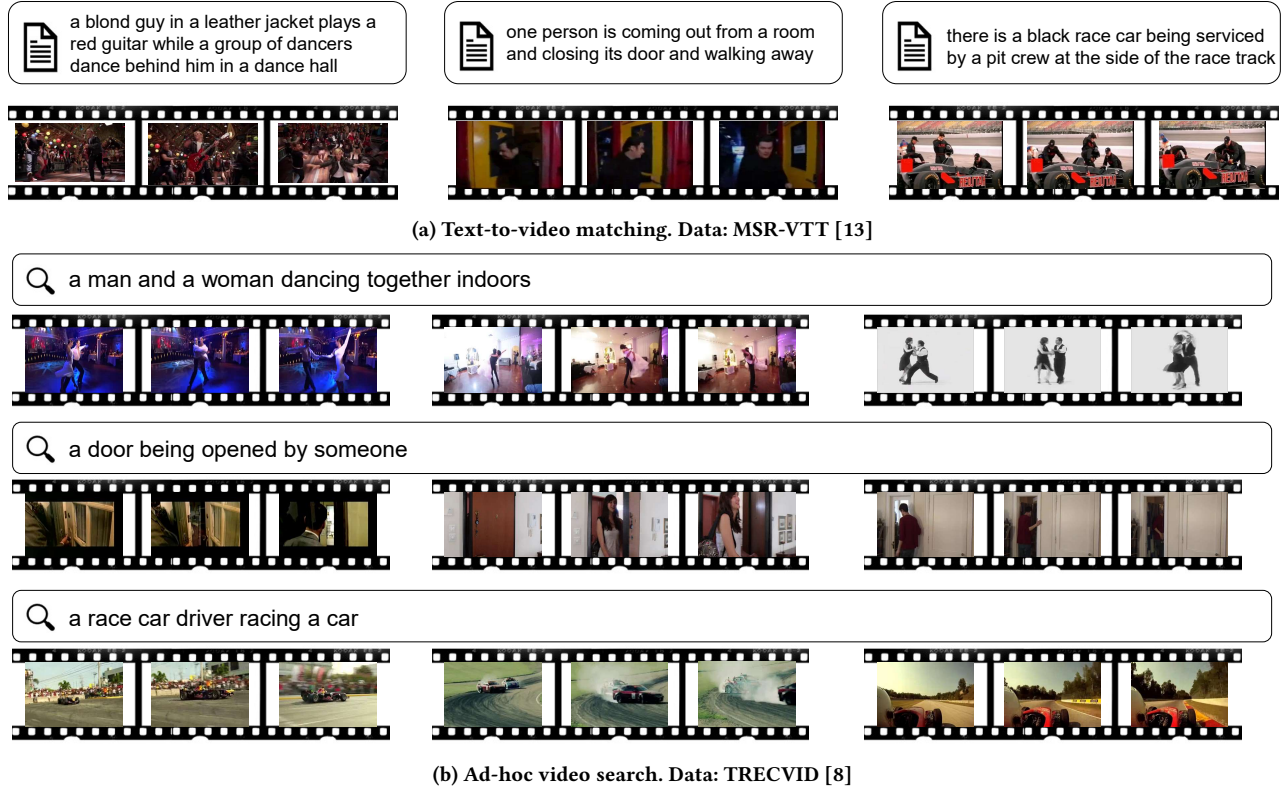
**Figure 1: Illustrating two types of text-to-video retrieval tasks: (a) text-to-video matching (TVM) and (b) ad-hoc video search (AVS). Compared to TVM which is to find a single match within thousands of videos, AVS aims to retrieve *multiple* relevant videos from a *million*-scale collection. This paper is targeted at the latter.**

matching [32, 35], aligning each text encoder with a unique common space [20, 24, 36], and generating multiple spaces that enable cross-matching between various text and video features [27, 37]. Nonetheless, these methods lack a specific design for enhancing the diversity of features across different spaces. Moreover, we believe that each feature, whether text or video, needs to learn its dominant space. And there is also a need for tailored training strategies that not only ensure accuracy but also enhance the diversity of retrieval results.

In this paper, we introduce a new network architecture, LPD (**L**earning **P**artially-**D**ecorrelated Common Spaces for Ad-hoc Video Search). The overall framework is depicted in Fig. 2a. To fully exploit the expressive capability of each feature within its corresponding space, we design the architecture so that one end of each space is connected to the corresponding feature, while the other end integrates a weighted fusion of features from the corresponding modality. To enhance the diversity of retrieval results across multiple spaces while ensuring relevance, we introduce a de-correlation loss that imposes constraints on the ordering of negative samples differently in each space. To enhance the consistency of multi-space convergence, we designed the entropy-based fair multi-space triplet ranking loss. By integrating the relevance-enhancing triplet ranking loss with the diversity-promoting de-correlation loss, LPD not only maintains the relevance of the retrieval outcomes but also boosts

their diversity across various spaces. What's more, de-correlation loss is model-agnostic and can be applied to other multi-space methods as well. In summary, our main contributions are as follows:

- To improve result diversity in the AVS, we propose a novel model LPD, which learns multiple feature-specific common spaces, exploiting the expressive capability of each feature.
- We design the de-correlation loss to enhance diversity and improve the multi-space triplet ranking loss by enforcing fair space convergence.
- Experiments on the long-standing TRECVID AVS benchmark series (2016–2023) verify the effectiveness of LPD. Source code is available at https://github.com/xxayt/LPD.

## 2 Related Work

### 2.1 Multi-feature Video Retrieval

Existing research on multi-feature video retrieval has developed various common-space construction styles. Single-space methods typically fuse multiple features into a robust composite feature for matching within a common space. For example, W2VV++ [33] and DE [30] integrate multiple features through vector concatenation, while CE [38] employs a collaborative experts model to aggregate features. Alternatively, some methods define several fixed common spaces that are not feature-specific. DualTask [34] constructs an

embedding space and a concept space. LAFF [39] uses a multi-head approach to establish multiple parallel common spaces. TMVM [40] constructs a multi-embedding space by learning diverse visual prototypes for each video. Further, some methods construct common spaces tailored to specific features. MMT [41] used a multimodal transformer to fuse multiple video features and create videofeature specific spaces. SEA [36] aligned each text feature with a unique common space. T×V [37] performed cross-matching for each text-video feature pair, resulting in a quadratic number of common spaces. Improved-ITV [42] employed generative captions for multi-stage fusion but ignores feature complementarity and space diversity. Our method differs by directly promoting diversity through space-wise regularization during training. In this paper, we propose LPD, which learns feature-specific spaces for all video and text features, coupled with a de-correlation loss designed to enhance the diversity across different spaces. The number of common spaces in LPD is equal to the number of features, making it more efficient than T×V.

## 2.2 Ad-hoc Video Search

Ad-hoc video search (AVS), evaluated annually in TRECVID, has been studied since 2016 [1]. In this task, given a textual topic description, the search system formulates a query and returns a ranked list of relevant video shots. Early efforts matched concept representations extracted from both videos and queries using predefined concept sets [43–49]. However, this required substantial manual effort for concept design and annotation [50, 51].

With advancements in deep neural networks and the availability of video captioning datasets, recent progress in ad-hoc video search typically employs visual-text embedding models [30, 33–37, 39, 52, 53]. Due to the need for comprehensive retrieval across diverse scenes in the AVS, teams typically employ deep models to extract multiple features and map them into one or more common spaces to calculate cross-modal similarity [19–26, 28, 29, 31, 32]. Some methods learn a single common space by fusing multiple features and enhance performance with additional training data and advanced features [19, 22, 31, 32]. Others design interpretable multi-space architectures for matching [21, 22], or construct multiple text-specific spaces and perform cross-matching [20, 23, 24, 27]. More recently, multi-model late fusion has also been explored [25–29, 54]. Beyond space construction, relevance-aware mining [55] and support-set training [56] have been proposed to better leverage unlabeled positives. Separately, general-purpose diversification methods such as MMR [57] and xQuAD [58] improve diversity in retrieval but may sacrifice relevance. In this paper, we construct diverse common spaces for multiple features and achieve a better trade-off without post-processing, making our approach more practical for AVS.

## 3 Proposed LPD Method

### 3.1 Overall Framework

We formalize an ad-hoc video search process as follows. We denote a specific video clip as $v$ and a large collection of $n$ *unlabeled* video clips as $\mathcal{V} = \{v_1, \ldots, v_n\}$. For an ad-hoc query in the form of a sentence $t$, let $cms(t, v)$ be a **c**ross-**m**odal **s**imilarity function that measures the semantic relevance between the query and a specific

video. Accordingly, the search process boils down to sorting $\mathcal{V}$ in descending order in terms of $cms(t, v)$ and returning the top-ranked items for the given query. The computation of $cms(t, v)$ requires proper embeddings of both $t$ and $v$ into a common cross-modal space.

For features, suppose all textual queries are represented by a set of $k_1$ sentence-level features $\{f_{t,1}, \ldots, f_{t,k_1}\}$, and the videos are represented by a set of $k_2$ video-level features, $\{f_{v,1}, \ldots, f_{v,k_2}\}$.

We need to establish a common space for each text feature and video feature, wherein we compute the similarity. As the features are obtained by distinct extractors and thus incompatible, we shall use a feature transformation layer to rectify the diverse features to the same length embeddings. Then, we could get sentence-level embeddings $\{e_{t,1}, \ldots, e_{t,k_1}\}$, and video-level embeddings, $\{e_{v,1}, \ldots, e_{v,k_2}\}$. Since each embedding independently dominates a common space, the cross-modal similarity $cms(t, v)$ is the average similarity across all common spaces, *i.e.*

$$
\begin{cases}
cms^i(t, v) = cosine[e_{t,i}, \text{Fusion}_{v,i}(e_{v,1}, \ldots, e_{v,k_2})], \\
cms^j(t, v) = cosine[\text{Fusion}_{t,j}(e_{t,1}, \ldots, e_{t,k_1}), e_{v,j}], \\
cms(t, v) = \frac{1}{k_1 + k_2} \left[ \sum_{i=1}^{k_1} cms^i(t, v) + \sum_{j=1}^{k_2} cms^j(t, v) \right].
\end{cases}
\tag{1}
$$

Where $cms^i(t, v)$ and $cms^j(t, v)$ represent the similarities of the common spaces dominated by the $i^{th}$ text feature and the $j^{th}$ video feature, respectively. Thus, for each common space, one end is the embedding obtained from an individual feature, and the other end is the fusion of embeddings from another modality. Different spaces aim to mine the information from various features, enhancing the diversity of retrieval and recalling more relevant videos.

Next, we describe in detail of feature transform and fusion, followed by the proposed de-correlation loss and fair multi-space triplet ranking loss.

### 3.2 Feature Transform and Fusion

Without loss of generality, we are provided with a diverse set of $k$ different features $\{f_1, \ldots, f_k\}$, sized as $d_1, \ldots, d_k$, respectively. As the features are obtained by distinct extractors and thus incompatible, we shall use a feature transformation layer to rectify the diverse features to be of the same length. To convert the $i$-th feature to a new $d$-dimensional feature embedding, we use

$$
e_i = \phi \left( \text{Linear}_{d_i \times d}(f_i) \right),
\tag{2}
$$

where $\phi$ is a nonlinear activation function. Following previous work [33, 36, 39], we use tanh as $\phi$.

Regarding the fusion method, as indicated in Eq. (1), each featurespecific common space is designed to match a specific feature with a dynamic weighted fusion of features from the opposite modality. Consequently, we contemplate utilizing a weighted fusion to integrate the features $e_1, \ldots, e_k$ from the other modality, where $k = k_1$ for text and $k = k_2$ for video

$$
\text{Fusion}(e_1, \ldots, e_k) = \sum_{i=1}^{k} a_i e_i,
\tag{3}
$$
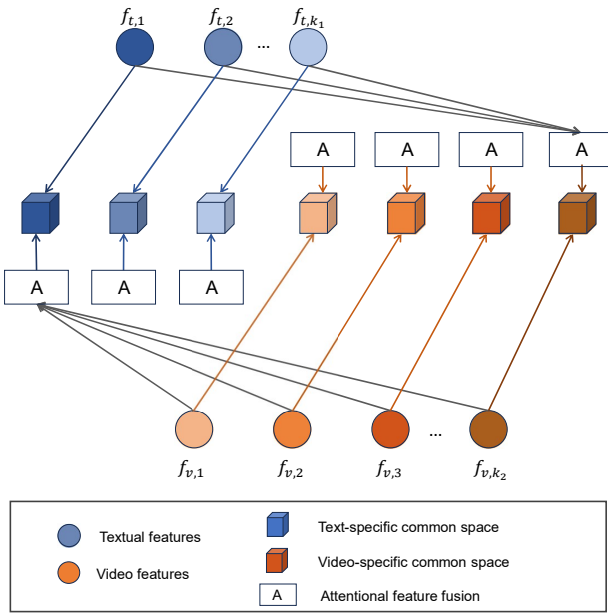
(a) Construction of multiple feature-specific common spaces
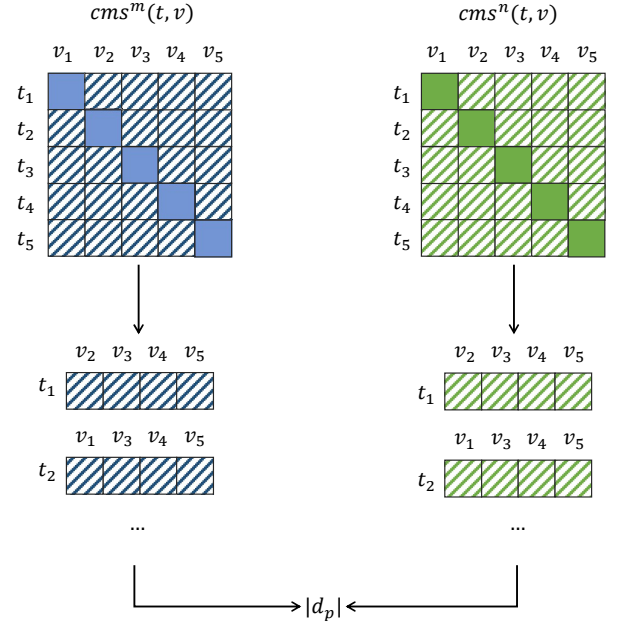
(b) Calculating proposed de-correlation loss for spaces $m$ and $n$

Figure 2: Proposed LPD. Each common space for video-text matching is feature-specific, with one end connected to its corresponding feature and the other end as a weighted fusion of features from the other modality. The feature transform layers associated with the same feature (indicated by the arrows originating from the feature in the figure) share parameters. For the de-correlation loss in a mini-batch with three video-text pairs, the blue and green squares represent the text-video similarity in spaces $m$ and $n$, respectively. Different rows correspond to different text queries, and different columns represent different videos. The goal is to improve the diversity of retrieval outcomes across various spaces by lowering Pearson correlation coefficient for negative videos across these spaces.

with weights $\{a_1, \ldots, a_k\}$ computed by a attention layer as:

$$\{a_1, \ldots, a_k\} = \text{softmax}\left(\text{Linear}_{d \times 1}(\{e_1, \ldots, e_k\})\right). \quad (4)$$

## 3.3 De-correlation Loss

To enhance the complementarity between the multiple common spaces while preserving their ability to retrieve positive samples, we propose an auxiliary loss that minimizes the correlation between the rankings of negative samples derived from the individual spaces. Since our method uses multi-embedding spaces, each space produces a ranking, and the final retrieval result is obtained by score-based fusion of these rankings, in Eq. (1). By minimizing the correlation between individual rankings, we naturally promote diversity in the final retrieval result, thereby enhancing its diversity.

As shown on Fig. 2b, for a specific space $m$, the $b \times b$ video-text similarity matrix is represented as $\text{cms}^m(t, v)$. Accordingly, the $i$ th row $\text{cms}^m(t_i, v)$ contains one positive video-text sample pair similarity $\text{cms}^m(t_i, v^+)$ and $b - 1$ negative video-text sample pair similarities $\text{cms}^m(t_i, v^-)$.

To apply the de-correlation loss, for each text in different spaces, we apply a constraint to increase the dispersion of these rankings among different spaces, thereby enhancing the diversity of retrieval outcomes. For two specific spaces (indexed by $m$ and $n$) from the

$(k_1 + k_2)$ available spaces, the de-correlation loss DcL is defined as:

$$\text{DcL} = \frac{1}{b} \sum_{i=1}^{b} \left| d_p \left(\text{cms}^m\left(t_i, v^-\right), \text{cms}^n\left(t_i, v^-\right)\right)\right|, \quad (5)$$

where $d_p$ is the Pearson correlation coefficient, which is applied between the ordered sets of negative samples across different spaces for a given text. We aim to optimize the absolute value of $d_p$ to 0 because our goal is to enhance differences in retrieval outcomes across different spaces, rather than making the retrieval outcomes completely opposite with $d_p = -1$.

To facilitate parallel processing of Eq. (5), we utilize a masking method for matrix operations on the video-text similarity matrices. Let $M$ be the similarity matrix for a specific space. The $i$-th row $M_i$ stores similarity scores of text $t_i$ to all videos in the batch. The diagonal elements represent positive sample pairs, while the off-diagonal elements represent negative sample pairs. In order to mask out the positive pairs when computing the row-wise rank correlation coefficient, we define a binary matrix $Z$, where $Z_{i,j} = 0$ for $i = j$ and $Z_{i,j} = 1$ otherwise. Applying element-wise multiplication with $Z$ on the similarity matrices w.r.t. space $m$ and $n$, DcL can be rewritten as:

$$\text{DcL} = \frac{1}{b} \sum_{i=1}^{b} \left| d_p \left((M^m \odot Z)_i, (M^n \odot Z)_i\right)\right|. \quad (6)$$

This allows for direct matrix operations for efficient computation of DcL and thus accelerates training.

## 3.4 Fair Multi-space Triplet Ranking Loss

During each training iteration, we sample a mini-batch B containing $b$ video-text pairs $\{(v_i, t_i) \mid i = 1, \ldots, b\}$ from the training dataset. This sampling strategy ensures that each video is irrelevant to other texts within the batch, and vice versa.

For a specific space $s$, we use the improved triplet ranking loss (ITRL) by Faghri et al. [59]. While originally proposed for image-text matching, ITRL is now found to be effective for text-video matching [30, 33, 36, 39, 60]. Unlike the classical triplet ranking loss that selects negative training examples by random, ITRL considers the negative that violates the ranking constraint the most (within a mini-batch) and is thus deemed to be the most informative for improving the model being trained. Given a training sentence $t$ with $v^+$ as a video relevant *w.r.t.* $t$ and $v^-$ as irrelevant, we express the ITRL of a specific space $s$ as

$$
\begin{cases}
v^{-*} & = \mathrm{argmax}_{v^- \in B} \left( \mathrm{cms}^s\left(t, v^-\right) - \mathrm{cms}^s\left(t, v^+\right) \right), \\
\mathrm{ITRL}^s & = \max\left( 0, \alpha + \mathrm{cms}^s\left(t, v^{-*}\right) - \mathrm{cms}^s\left(t, v^+\right) \right),
\end{cases} \tag{7}
$$

where $\alpha$ is a positive hyper-parameter concerning the margin.

However, applying ITRL to multiple spaces presents a challenge. Due to the diversity across different spaces, convergence rates often vary, and directly summing the ITRL from different spaces may lead to overfitting in some spaces. This phenomenon arises because different features, which dominate different spaces, exhibit considerable variations in dimensions, representational capabilities for different modal information, and update frequencies, leading to disparate convergence speeds among spaces. If a space has converged, subsequent training is likely to result in overfitting [61, 62].

Intuitively, spaces with slower convergence rates, wherein feature distribution updates less frequently, require more extensive training. Inspired by the entropy of feature [63] and One-shot Supernet [64], we propose a multi-space triplet ranking loss with fair space convergence. It dynamically selects different spaces for training based on the information entropy of features, promoting a more balanced and effective training process across all spaces and leading to more stable convergence. Next, we detail the method for calculating space weights based on feature information entropy and the process for computing an improved multi-space loss.

*3.4.1 Feature Entropy based Common Space Weighting.* To compute the feature information entropy for space $s$, we collect all its features in the given batch and apply min-max normalization independently to each dimension. This yields a set of $N = b \times d$ values in the range $[0, 1]$. These values are then quantized into 100 bins, converting them into a 100-dimensional probability distribution $P_s$. The entropy $h_s$ is computed as:

$$
h_s = -\sum_{j=1}^{100} P_{s,j} \log(P_{s,j} + \epsilon), \tag{8}
$$

where $\epsilon$ is a small positive constant ensuring numerical stability. Given the $k_1 + k_2$ distinct spaces, we obtain a space-wise entropy vector $H = \{h_1, h_2, \ldots, h_{k_1+k_2}\}$.

A space-wise weight vector, denoted as $W = \{w_1, w_2, \ldots, w_{k_1+k_2}\}$, is computed based on $H$ as:

$$
W = \mathrm{softmax}(\tanh(H)). \tag{9}
$$

*3.4.2 Multi-space loss.* Once the space weight vector $W$ is dynamically given per iteration, each space is either included for or excluded from training, depending on whether its weight exceeds a given threshold (which is $\frac{1}{k_1+k_2}$). The final multi-space loss function, denoted as EF-MTRL (Entropy-based Fair Multi-space Triplet Ranking Loss) is computed as:

$$
\mathrm{EF\text{-}MTRL} = \sum_{s=1}^{k_1+k_2} \left[ w_s > \frac{1}{k_1 + k_2} \right] \cdot \mathrm{ITRL}^s. \tag{10}
$$

## 3.5 Overall Loss

The overall loss function is simply the sum of DcL and EF-MTRL. By integrating these two individual losses, our model is motivated to not only maintain the relevance of the retrieval outcomes but also to boost their diversity across various spaces. This balanced approach tackles the dual challenge of ensuring both relevance and diversity in AVS, leading to a more effective and encompassing search performance.

## 4 Evaluation

### 4.1 Experimental Setup

*4.1.1 Test data.* We adopt the TRECVID evaluation [1], the *de facto* international benchmark for AVS. Since the AVS 2024 data has not been released, we conduct experiments on the TV16-23 benchmark series, see Tab. 1.

**Table 1: Test sets used by the TRECVID (TV) AVS benchmark series. Frames are obtained by uniform sampling with a fixed time interval of 0.5 seconds.**

| Test set | Video clips | Frames | Test queries | |
|----------|-------------|--------|--------------|------|
| IACC.3 [1] | 335,944 | 3,845,221 | TV16 | 30 |
| | | | TV17 | 30 |
| | | | TV18 | 30 |
| V3C1 [8] | 1,082,649 | 7,839,450 | TV19 | 30 |
| | | | TV20 | 20 |
| | | | TV21 | 20 |
| V3C2 [11] | 1,425,443 | 6,082,291 | TV22 | 30 |
| | | | TV23 | 20 |

*4.1.2 Development data.* For training, we adopt the 9K subset of MSR-VTT [65]. Following [19, 31], we adopt the TV16 video-text matching test set of 200 video-text pairs (TV16-VTT) as our validation set for model selection.

*4.1.3 Performance metric.* We follow the TRECVID protocol, reporting inferred Average Precision (infAP) [1].

*4.1.4 Test of significance.* We perform a randomization test to check if the performance difference between two video retrieval systems is statistically significant [66].

**Table 2: Video/text features used in our evaluation. Video-level features are obtained by mean pooling over frames or segments unless otherwise stated.**

| Feature | Dim. | Short description |
|---|---|---|
| *Video features:* | | |
| wsl | 2,048 | ResNeXt-101 pre-trained on 940 million public images, fine-tuned on ImageNet1k [67]. |
| clip-b32 | 512 | CLIP (ViT-B/32) pre-trained on web-scale image-text pairs by contrastive learning [15]. |
| clip-l14 | 768 | CLIP (ViT-L/14), with a larger vision transformer. |
| ircsn | 2,048 | irCSN-152 trained by weakly supervised learning on IG-65M [68]. |
| beit | 2,048 | BEiT pre-trained on full ImageNet and fine-tune on 1k-class ImageNet [69]. |
| blip | 256 | BLIP(ViT-B) pre-trained on 129M image-text pairs [70]. |
| *Text features:* | | |
| blip | 256 | Text transformer from BLIP(ViT-B). |
| clip-b32 | 512 | Text transformer from CLIP(ViT-B/32). |
| clip-l14 | 768 | Text transformer from CLIP(ViT-L/14). |

*4.1.5 Implementations.* The networks have an output dimension of $d = 512$. The margin $\alpha$ in Eq. (7) is empirically set to 0.2. Training is performed using SGD with a mini-batch size of 128, utilizing RMSProp as the optimizer. The initial learning rate is $10^{-4}$, reduced by a factor of 0.99 after each epoch. We adopt mean Average Precision (mAP) for model selection: early stop occurs once there is no improvement in mAP on the validation set for ten consecutive epochs. We use six video features and three text features (Tab. 2). All experiments are conducted with four 3090 GPUs.

## 4.2 Comparison with Baselines

*4.2.1 Baselines.* To ensure a fair and reproducible comparison, we select baselines with the following three criteria: 1) open source, 2) bi-encoder style for million-scale video retrieval as AVS, and 3) peer reviewed. As such, we obtain 15 baselines in four groups:
• Group I (zero-shot retrieval): Directly match the text embedding with the average pooled embedding of all frames for CLIP and BLIP.
• Group II (CLIP-based end-to-end networks): CLIP [15], CLIP4Clip [4], X-CLIP [3], TS2-Net [5], CLIP-VIP [16], DGL [71], and Teach-CLIP [72].
• Group III (Multi-feature fusion, single space learning): DE [30] and W2VV++ [33].
• Group IV (Multi-feature fusion, multi space learning): DualTask [34], SEA [36], LAFF [39], and T×V [37].

*4.2.2 Results.* Tab. 3 presents a comprehensive comparison among various models across three benchmark datasets, IACC.3, V3C1 and V3C2, evaluated over different years from TV16 to TV23. We observe that models integrating multiple features generally surpass those relying on a single feature CLIP/BLIP in performance. Even the best-performing blip zero-shot retrieval method significantly lags behind multi-feature approaches. This trend underscores the advantage of leveraging diverse feature sets, which, in combination,

offer a richer representation of video and text, thereby enhancing retrieval precision. Particularly noteworthy is the performance of models classified under "Learning multiple common spaces for multiple features" which includes DualTask, LAFF, SEA, T×V and our proposed LPD. These models, by dedicating unique spaces for individual features, facilitate a more subtle understanding and utilization of each feature's distinct characteristics.

Comparing LPD to other multi-space baselines in Tables 4 and 5, we observe a significant improvement. This advancement can be attributed to two key factors. Firstly, LPD is designed to train a distinct space for each feature, complemented by a dynamic weighted fusion of features from the opposite modality. DualTask and LAFF manually set multiple spaces, while SEA focuses on text features, and T×V performs cross-matching between text and video features. These methods do not fully leverage the potential of feature-specific space optimization. Secondly, LPD employs a de-correlation loss during training to constrain the ordering of negative samples across different spaces. This enhancement foregrounds the distinctiveness of each space, further elevating the retrieval performance.

## 4.3 Understanding LPD

*4.3.1 Inter-space Divergence.* In order to measure how each common space differs from another, we compute an inter-space similarity as follows. Given $S_i$ and $S_j$ as top-20 retrieved videos from the $i$-th space and the $j$-th space *w.r.t.* the TV23 queries respectively, the inter-space similarity is computed as the Intersection over Union (IoU) between $S_i$ and $S_j$. The IoU heatmaps of LAFF, LPD *w/o* DcL, and LPD are shown in Fig. 3.
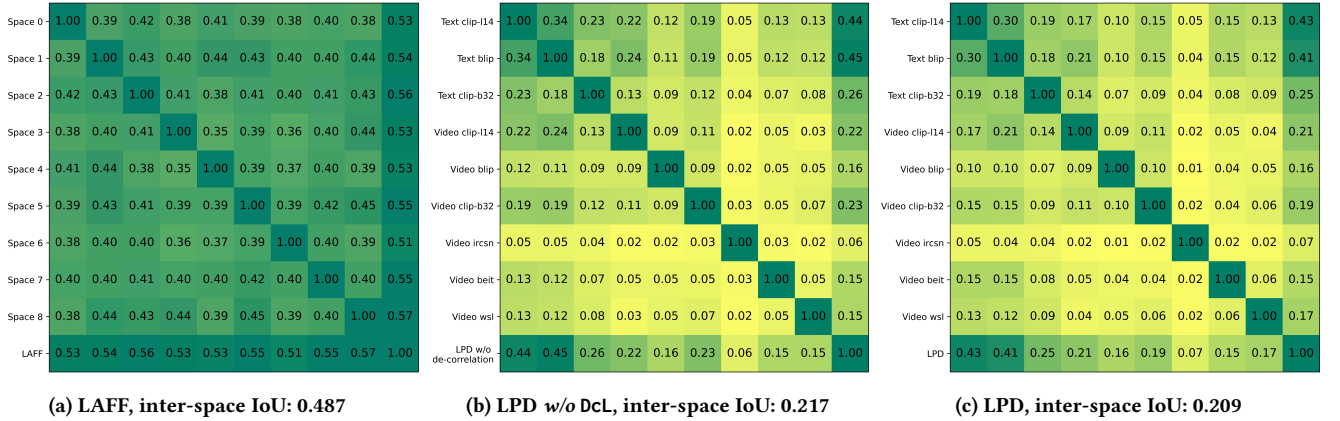
For the LAFF model, using a multi-head approach to construct nine parallel common spaces, results in a lack of clear distinction between each space. As shown in Fig. 3a, the high similarity between different spaces, indicated by large green areas in the heatmap, suggests that the LAFF model's spaces lack diversity. Comparing Fig. 3b with Fig. 3a, we observed a notable increase in the distinctiveness among the spaces, highlighted by the overall shift to yellow in the heatmap. This shift verifies that our feature-specific design successfully created diversity within the common spaces. Further, when comparing Fig. 3c to Fig. 3b, we observed a significant decrease in the IoU of various spaces (for example, the IoU between text clip-14 and text blip decreased from 0.34 to 0.30), indicating the effectiveness of our proposed de-correlation loss in increasing the diversity among the common spaces.

*4.3.2 Partial versus full de-correlation loss.* As shown in Fig. 2b, we only constrain *negative* examples. Here, we provide experimental evidence to support this design choice. In Tab. 4, we compare different types of DcL settings. We observe that compared to the LPD without de-correlation loss, LPD with full de-correlation loss shows a slight decrease in performance (MEAN infAP $0.229 \rightarrow 0.227$), whereas LPD (partial de-correlation loss) improves the MEAN infAP to 0.241. This indicates that applying de-correlation loss to a full list that includes *positive* examples consequently impairs retrieval performance. The reason is that applying de-correlation to the *Full* ranking list would disrupt the correct ranking of the positive training samples, and consequently result in performance loss. By constraining only the correlation of *negative* training examples,

**Table 3: Comparison with baselines. All the multi-feature methods (Group III and IV) use the same set of video / text features (Tab. 2). LPD outperforms other methods significantly (p<0.05).**

| Method | IACC.3 | | | V3C1 | | | V3C2 | | MEAN |
|---|---|---|---|---|---|---|---|---|---|
| | TV16 | TV17 | TV18 | TV19 | TV20 | TV21 | TV22 | TV23 | |
| *Zero-shot retrieval*: | | | | | | | | | |
| clip-b32 | 0.173 | 0.208 | 0.087 | 0.136 | 0.161 | 0.194 | 0.119 | 0.102 | 0.148 |
| clip-l14 | 0.201 | 0.232 | 0.103 | 0.082 | 0.102 | 0.169 | 0.118 | 0.116 | 0.140 |
| blip | 0.196 | 0.217 | 0.121 | 0.159 | 0.203 | 0.223 | 0.144 | 0.157 | 0.178 |
| *CLIP-based end-to-end networks*: | | | | | | | | | |
| CLIP4Clip [4] | 0.196 | 0.228 | 0.108 | 0.142 | 0.161 | 0.183 | 0.127 | 0.139 | 0.161 |
| X-CLIP [3] | 0.209 | 0.229 | 0.114 | 0.150 | 0.184 | 0.195 | 0.125 | 0.129 | 0.167 |
| TS2-Net [5] | 0.202 | 0.266 | 0.132 | 0.120 | 0.153 | 0.188 | 0.150 | 0.133 | 0.168 |
| CLIP-VIP [16] | 0.184 | 0.212 | 0.109 | 0.143 | 0.148 | 0.175 | 0.109 | 0.088 | 0.146 |
| DGL [71] | 0.194 | 0.224 | 0.121 | 0.139 | 0.136 | 0.159 | 0.110 | 0.111 | 0.149 |
| TeachCLIP [72] | 0.199 | 0.248 | 0.129 | 0.176 | 0.196 | 0.218 | 0.166 | 0.164 | 0.187 |
| *Multi-feature, single-space*: | | | | | | | | | |
| W2VV++ [33] | 0.215 | 0.301 | 0.138 | 0.189 | 0.226 | 0.213 | 0.177 | 0.186 | 0.206 |
| DE [30] | 0.204 | 0.319 | 0.138 | 0.189 | 0.235 | 0.225 | 0.172 | 0.200 | 0.210 |
| *Multi-feature, multi-space*: | | | | | | | | | |
| DualTask [34] | 0.209 | 0.332 | 0.144 | 0.192 | 0.240 | 0.224 | 0.178 | 0.203 | 0.215 |
| LAFF [39] | 0.222 | 0.287 | 0.142 | 0.192 | 0.225 | 0.237 | 0.175 | 0.183 | 0.208 |
| SEA [36] | 0.231 | 0.314 | 0.158 | 0.199 | 0.251 | 0.230 | 0.188 | 0.220 | 0.224 |
| T × V [37] | 0.240 | 0.313 | 0.159 | 0.211 | 0.243 | 0.260 | 0.198 | 0.200 | 0.228 |
| LPD | **0.245** | **0.327** | **0.169** | **0.218** | **0.287** | **0.269** | **0.221** | **0.221** | **0.245** |



(a) LAFF, inter-space IoU: 0.487   (b) LPD *w/o* `DcL`, inter-space IoU: 0.217   (c) LPD, inter-space IoU: 0.209

**Figure 3: Visualizing inter-space similarities for LAFF, LPD *w/o* `DcL`, and LPD. Per model, we calculate the Intersection over Union (IoU) of the top-20 retrieval results by the individual spaces. Lower IoU scores indicate larger inter-space diversity. Best viewed in color.**

the proposed de-correlation loss indirectly yet effectively strikes a balance between search result diversity and relevance.

*4.3.3 `DcL` improves baseline models.* De-correlation loss is designed to be model-agnostic, with its primary goal being to diversify the

ordering of negative samples across different common spaces. In Tab. 5, we integrate `DcL` into three multi-space baselines: LAFF, SEA, and T×V. The results reveal that incorporating de-correlation loss into these models led to an approximate 3% improvement in mean infAP scores. This improvement strongly indicates that

**Table 4: Ablation study. LPD (Partial De-correlation) achieves significant improvement (p< 0.05).**

| Setup | IACC.3 | | | V3C1 | | | V3C2 | | MEAN |
|---|---|---|---|---|---|---|---|---|---|
| | TV16 | TV17 | TV18 | TV19 | TV20 | TV21 | TV22 | TV23 | |
| LPD (Partial DcL) | **0.245** | **0.327** | **0.169** | **0.218** | **0.287** | **0.269** | **0.221** | **0.221** | **0.245** |
| Partial DcL → Full DcL | 0.230 | 0.309 | 0.146 | 0.203 | 0.265 | 0.251 | 0.201 | 0.213 | 0.227 |
| *w/o* DcL | 0.235 | 0.321 | 0.150 | 0.207 | 0.262 | 0.261 | 0.199 | 0.197 | 0.229 |
| *w/o* EF-MTRL | 0.244 | 0.325 | 0.165 | 0.216 | 0.278 | **0.269** | 0.215 | 0.220 | 0.241 |

**Table 5: Performance enhancement from de-correlation loss on multi-space baseline models. Adding DcL to individual models consistently brings in significant improvement (p<0.05).**

| Model | IACC.3 | | | V3C1 | | | V3C2 | | MEAN |
|---|---|---|---|---|---|---|---|---|---|
| | TV16 | TV17 | TV18 | TV19 | TV20 | TV21 | TV22 | TV23 | |
| LAFF | 0.222 | 0.287 | 0.142 | 0.192 | 0.225 | 0.237 | 0.175 | 0.183 | 0.208 |
| LAFF+DcL | 0.229 | 0.296 | 0.139 | 0.198 | 0.247 | 0.230 | 0.183 | 0.194 | 0.215 (+3.4%↑) |
| SEA | 0.231 | 0.314 | 0.158 | 0.199 | 0.251 | 0.230 | 0.188 | 0.220 | 0.224 |
| SEA+DcL | 0.229 | 0.337 | 0.157 | 0.200 | 0.258 | 0.249 | 0.200 | 0.209 | 0.230 (+2.7%↑) |
| T×V | 0.240 | 0.313 | 0.159 | 0.211 | 0.243 | 0.260 | 0.198 | 0.200 | 0.228 |
| T×V+DcL | 0.241 | 0.319 | 0.159 | 0.212 | 0.259 | 0.266 | 0.207 | 0.207 | 0.234 (+2.6%↑) |



(a) LPD *w/o* EF-MTRL

(b) LPD *w/* EF-MTRL
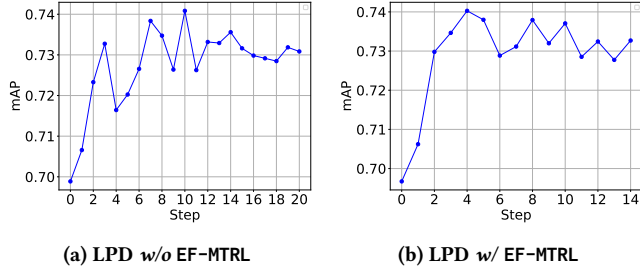
**Figure 4: Changes in mAP on the validation set. The EF-MTRL method enables faster convergence.**

de-correlation loss is not only effective but also highly adaptable across different model architectures. This adaptability confirms de-correlation loss as a powerful tool for increasing the diversity and accuracy of video retrieval systems in ad-hoc search tasks, demonstrating its broad applicability and effectiveness.

*4.3.4 The effect of EF-MTRL.* Fig. 4 shows the changes in mAP on the validation set, clearly indicating that the model with EF-MTRL converges faster and more stably compared to the one without it. As shown in Tab. 4, removing EF-MTRL (*w/o* EF-MTRL) and using the standard triplet ranking loss causes a slight drop in infAP from 0.245 to 0.241. This suggests that enforcing fair convergence across spaces enhances model effectiveness through more balanced learning and optimization among diverse feature spaces.

We observe that different feature spaces converge at different speeds during training, with some having a much larger impact on the overall loss. Our method gradually balances these differences,

leading to more synchronized convergence. This suggests that EF-MTRL acts as a regularizer by encouraging greater training stability and preventing overfitting to any single feature space.

*4.3.5 Complexity analysis.* As shown in Tab. 6, compared to LAFF, LPD does not increase the number of parameters or FLOPs.

**Table 6: Complexity analysis of *multi-feature, multi-space* methods.** $D$: total dimension of input features. FLOPs are estimated per video-text matching.

| Method | Trainable Params. | FLOPs (M) |
|---|---|---|
| T×V | $4 \times D \times d$ | 75.50 |
| DualTask | $D \times d + 11{,}147 \times d$ | 41.70 |
| LAFF | $D \times d + 9 \times d$ | 18.89 |
| LPD | $D \times d + 9 \times d$ | 18.89 |
| SEA | $D \times d$ | 18.87 |

## 5 Summary and Conclusions

We have developed LPD that enhances both relevance and diversity for ad-hoc video search (AVS) at a million-video scale. LPD learns diverse feature-specific common spaces and dynamically fuses information across modalities. A de-correlation loss promotes diversity in negative sample ranking, while an entropy-based multi-space triplet loss ensures relevance. Evaluations on the TRECVID AVS benchmarks (2016–2023) show that LPD outperforms a number of competitive baselines, setting a new standard for diverse and scalable video retrieval.

# References

[1] George Awad, Fiscus Jonathan, Joy David, Michel Martial, Smeaton Alan, Kraaij Wessel, Quenot Georges, Eskevich Maria, Aly Robin, Ordelman Roeland, Jones Gareth, Huet Benoit, and LarsonMartha. 2016. TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization, and Hyperlinking. In *TRECVID*.

[2] Chen Jiang, Hong Liu, Xuzheng Yu, Qing Wang, Yuan Cheng, Jia Xu, Zhongyi Liu, Qingpei Guo, Wei Chu, Ming Yang, et al. 2023. Dual-Modal Attention-Enhanced Text-Video Retrieval with Triplet Partial Margin Contrastive Learning. In *ACM MM*.

[3] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. 2022. X-CLIP: End-to-End Multi-grained Contrastive Learning for Video-Text Retrieval. In *ACM MM*.

[4] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.

[5] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. 2022. TS2-Net: Token shift and selection transformer for text-video retrieval. In *ECCV*.

[6] George Awad, Asad Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Willie McClinton, Martial Michel, Alan Smeaton, Yvette Graham, Wessel Kraaij, Georges Quenot, Maria Eskevich, Roeland Ordelman, Gareth Jones, and Benoit Huet. 2018. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *TRECVID*.

[7] George Awad, Asad Gov, Asad Butt, Keith Curtis, Yooyoung Lee, yooyoung@nist Gov, Jonathan Fiscus, David Joy, Andrew Delgado, Alan Smeaton, Yvette Graham, Wessel Kraaij, Georges Quenot, Joao Magalhaes, and Saverio Blasi. 2018. TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search. In *TRECVID*.

[8] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Godil Afzal, Andrew Delgado, Zhang Jesse, Eliot Godard, Lukas Diduch, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quenot. 2019. TRECVID 2019: An evaluation campaign to benchmark Video Activity Detection, Video Captioning and Matching, and Video Search and retrieval. In *TRECVID*.

[9] George Awad, Asad A Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, et al. 2020. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *TRECVID*.

[10] George Awad, Asad A Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, et al. 2021. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *TRECVID*.

[11] George Awad, Keith Curtis, Asad Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Eliot Godard, Lukas Diduch, Jeffrey Liu, et al. 2022. An overview on the evaluated video retrieval tasks at TRECVID 2022. In *TRECVID*.

[12] George Awad, Keith Curtis, Asad Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Eliot Godard, Lukas Diduch, Deepak Gupta, Dina Demner Fushman, Yvette Graham, Georges Quénot, et al. 2023. TRECVID 2023 - A series of evaluation tracks in video understanding. In *TRECVID*.

[13] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*.

[14] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3C1 Dataset: An Evaluation of Content Characteristics. In *ICMR*.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

[16] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. 2023. CLIP-ViP: Adapting Pre-trained Image-Text Model to Video-Language Alignment. In *ICLR*.

[17] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. 2023. Cap4video: What can auxiliary captions do for text-video retrieval?. In *CVPR*.

[18] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. 2023. Diffusionret: Generative text-video retrieval with diffusion model. In *ICCV*.

[19] Xirong Li, Jinde Ye, Chaoxi Xu, Shanjinwen Yun, Leimin Zhang, Xun Wang, Rui Qian, and Jianfeng Dong. 2019. Renmin University of China and Zhejiang Gongshang University at TRECVID 2019: Learn to Search and Describe Videos. In *TRECVID*.

[20] Xirong Li, Fangming Zhou, and Aozhu Chen. 2020. Renmin University of China at TRECVID 2020: Sentence Encoder Assembly for Ad-hoc Video Search. In *TRECVID*.

[21] Jiaxin Wu, Phuong Anh Nguyen, and Chong-Wah Ngo. 2020. VIREO@ TRECVID 2020 Ad-hoc Video Search. In *TRECVID*.

[22] Jiaxin Wu, Phuong Anh Nguyen, and Chong-Wah Ngo. 2021. VIREO@ TRECVID 2021 ad-hoc video search. In *TRECVID*.

[23] Fangming Zhou, Yihui Shi, Changqiao Wu, Xiaofeng Guo, Haofan Wang, Jincan Deng, and Debing Zhang. 2021. Kuaishou at TRECVID 2021: Two-stage Ranking Strategy for Ad-hoc Video Search. In *TRECVID*.

[24] Xirong Li, Aozhu Chen, Fan Hu, Xinru Chen, Chengbo Dong, and Gang Yang. 2021. Renmin University of China at TRECVID 2021: Searching and Describing Video. In *TRECVID*.

[25] Kazuya Ueki, Yuma Suzuki, Hiroki Takushima, Hideaki Okamoto, Hayato Tanoue, and Takayuki Hori. 2022. Waseda meisei softbank at TRECVID 2022. In *TRECVID*.

[26] Xirong Li, Aozhu Chen, Ziyue Wang, Fan Hu, Kaibin Tian, Xinru Chen, and Chengbo Dong. 2022. Renmin University of China at TRECVID 2022: Improving Video Search by Feature Fusion and Negation Understanding. In *TRECVID*.

[27] Konstantinos Gkountakos, Damianos Galanopoulos, Despoina Touska, Konstantinos Ioannidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris. 2022. ITI-CERTH participation in ActEV and AVS tracks of TRECVID 2022. In *TRECVID*.

[28] Kazuya Ueki, Yuma Suzuki, Hiroki Takushima, Haruki Sato, Takumi Takada, Hideaki Okamoto, Hayato Tanoue, Takayuki Hori, and Aiswariya Manoj Kumar. 2023. Waseda Meisei SoftBank at TRECVID 2023. In *TRECVID*.

[29] Xirong Li, Fan Hu, Ruixiang Zhao, Ziyuan Wang, Jingyu Liu, Jiazhen Liu, Bangxiang Lan, Wenguan Kou, Yuhan Fu, and Zhanhui Kang. 2023. Renmin University of China and Tencent at TRECVID 2023: Harnessing Pre-trained Models for Ad-hoc Video Search. In *TRECVID*.

[30] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *CVPR*.

[31] Xirong Li, Jianfeng Dong, Chaoxi Xu, Jing Cao, Xun Wang, and Gang Yang. 2018. Renmin University of China and Zhejiang Gongshang University at TRECVID 2018: Deep Cross-Modal Embeddings for Video-Text Retrieval. In *TRECVID*.

[32] Yida Zhao, Yuqing Song, Shizhe Chen, and Qin Jin. 2020. RUC_AIM3 at TRECVID 2020: Ad-hoc Video Search & Video to Text Description. In *TRECVID*.

[33] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2VV++: Fully deep learning for ad-hoc video search. In *ACM MM*.

[34] Jiaxin Wu and Chong-Wah Ngo. 2020. Interpretable embedding for ad-hoc video search. In *ACM MM*.

[35] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-Grained Video-Text Retrieval With Hierarchical Graph Reasoning. In *CVPR*.

[36] Xirong Li, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, and Gang Yang. 2021. SEA: Sentence encoder assembly for video retrieval by textual queries. *TMM* 23 (2021), 4351–4362.

[37] Damianos Galanopoulos and Vasileios Mezaris. 2022. Are All Combinations Equal? Combining Textual and Visual Features with Multiple Space Learning for Text-Based Video Retrieval. In *ECCV Workshops*.

[38] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use What You Have: Video retrieval using representations from collaborative experts. In *BMVC*.

[39] Fan Hu, Aozhu Chen, Ziyue Wang, Fangming Zhou, Jianfeng Dong, and Xirong Li. 2022. Lightweight Attentional Feature Fusion: A New Baseline for Text-to-Video Retrieval. In *ECCV*.

[40] Chengzhi Lin, Ancong Wu, Junwei Liang, Jun Zhang, Wenhang Ge, Wei-Shi Zheng, and Chunhua Shen. 2022. Text-adaptive multiple visual prototype matching for video-text retrieval. *NeurIPS*.

[41] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multimodal transformer for video retrieval. In *ECCV*.

[42] Jiaxin Wu, Chong-Wah Ngo, and Wing-Kwong Chan. 2024. Improving Interpretable Embeddings for Ad-hoc Video Search with Generative Captions and Multi-word Concept Bank. In *ICMR*.

[43] Cees G. M. Snoek and Marcel Worring. 2009. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval* 2 (2009), 215–322.

[44] Milind Naphade, J.R. Smith, Jelena Tesic, S. Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. 2006. Large-Scale Concept Ontology for Multimedia. *IEEE Transactions on Multimedia* 13 (08 2006), 86–91.

[45] Yu-Gang Jiang, Jun Yang, Chong-Wah Ngo, and Alexander Hauptmann. 2010. Representations of Keypoint-Based Semantic Concept Detection: A Comprehensive Study. *IEEE Transactions on Multimedia* 12 (02 2010), 42–53.

[46] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. 2006. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *ACM MM*.

[47] Vinh-Tiep Nguyen, Duy-Dinh Le, Benjamin Renoust, Thanh Duc Ngo, Minh-Triet Tran, Duc Anh Duong, and Shinichi Satoh. 2016. NII-HITACHI-UIT at TRECVID 2016 Ad-hoc Video Search: Enriching Semantic Features using Multiple Neural Networks. In *TRECVID*.

[48] Kazuya Ueki, Yu Nakagome, Koji Hirakawa, Kotaro Kikuchi, Yoshihiko Hayashi, Tetsuji Ogawa, and Tetsunori Kobayashi. 2018. Waseda Meisei at TRECVID 2018:Ad-hoc Video Search. In *TRECVID*.

[49] Po-Yao Huang, Junwei Liang, Vaibhav, Xiaojun Chang, and Alexander Hauptmann. 2018. Informedia@TRECVID 2018:Ad-hoc Video Search with Discrete and Continuous Representations. In *TRECVID*.

[50] Phuong Anh Nguyen, Yi-Jie Lu, Hao Zhang, and Chong-Wah Ngo. 2018. Enhanced VIREO KIS at VBS 2018. In *MMM*.

[51] Takayuki Ueki, Kazuya an Hori and Tetsunori Kobayashi. 2019. Waseda_Meisei_SoftBank at TRECVID 2019: Ad-hoc Video Search. In *TRECVID*.

[52] Amirhossein Habibian, Thomas Mensink, and Cees G. M. Snoek. 2014. VideoStory: A New Multimedia Embedding for Few-Example Recognition and Translation of Events. In *ACM MM*.

[53] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidlere. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*.

[54] Jiangshan He, Ruizhe Li, Jiahao Guo, Hong Zhang, Mingxi Li, Zhengqian Wu, Zhongyuan Wang, Bo Du, and Chao Liang. 2023. WHU-NERCMS at TRECVID 2023: Ad-hoc Video Search (AVS) and Deep Video Understanding (DVU) Tasks. In *TRECVID*.

[55] Alex Falcon, Giuseppe Serra, and Oswald Lanz. 2024. Improving semantic video retrieval models by training with a relevance-aware online mining strategy. *CVUI* 245 (2024), 104035.

[56] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. 2021. Support-set bottlenecks for video-text representation learning. In *ICLR*.

[57] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR*.

[58] Rodrygo L.T. Santos. 2012. Explicit web search result diversification. *SIGIR Forum* 47 (2012), 67–68.

[59] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*.

[60] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. 2018. Learning Joint Embedding with Multimodal Cues for Cross-Modal Video-Text Retrieval. In *ICMR*.

[61] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An introduction to statistical learning*. Springer.

[62] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[63] Xu Ma, Pengjie Wang, Hui Zhao, Shaoguo Liu, Chuhan Zhao, Wei Lin, Kuang-Chih Lee, Jian Xu, and Bo Zheng. 2021. Towards a Better Tradeoff between Effectiveness and Efficiency in Pre-Ranking: A Learnable Feature Selection based Approach. In *SIGIR*.

[64] He Wei, Yuekui Yang, Haiyang Wu, Yangyang Tang, Meixi Liu, and Jianfeng Li. 2023. Automatic Feature Selection By One-Shot Neural Architecture Search In Recommendation Systems. In *WWW*.

[65] Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *ECCV*.

[66] Jianfeng Dong, Xirong Li, and Duanqing Xu. 2018. Cross-Media Similarity Evaluation for Web Image Retrieval in the Wild. *TMM* 20, 9 (2018), 2371–2384.

[67] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the Limits of Weakly Supervised Pretraining. In *ECCV*.

[68] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. 2019. Large-Scale Weakly-Supervised Pre-training for Video Action Recognition. In *CVPR*.

[69] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*.

[70] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*.

[71] Xiangpeng Yang, Linchao Zhu, Xiaohan Wang, and Yi Yang. 2024. DGL: Dynamic Global-Local Prompt Tuning for Text-Video Retrieval. In *AAAI*.

[72] Kaibin Tian, Ruixiang Zhao, Zijie Xin, Bangxiang Lan, and Xirong Li. 2024. Holistic Features are almost Sufficient for Text-to-Video Retrieval. In *CVPR*.