Self-supervised Video Summarization Guided by Semantic Inverse Optimal Transport

Yutong Wang
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China

Hongteng Xu
Gaoling School of Artifical Intelligence
Renmin University of China
Beijing, China

Dixin Luo*
School of Computer Science and Technology
Beijing Institute of Technology
Beijing, China

ABSTRACT

Video summarization is a critical task in video analysis that aims to create a brief yet informative summary of the original video (i.e., a set of keyframes) while retaining its primary content. Supervised summarization methods rely on time-consuming keyframe labeling and thus often suffer from the insufficiency issue of training data. In contrast, the performance of unsupervised summarization methods is often unsatisfactory due to the lack of semantically-meaningful guidance on the keyframe selection. In this study, we propose a novel self-supervised video summarization framework with the help of computational optimal transport techniques. Specifically, we generate textual descriptions from video shots and learn the projection from the textual embeddings to the visual ones together with an optimal transport plan between them via solving an inverse optimal transport problem. We propose an alternating optimization algorithm to solve this problem efficiently and design an effective mechanism in the algorithm to avoid trivial solutions. Given the optimal transport plan and the underlying distance between the projected textual embeddings and the visual ones, we synthesize pseudo-significance scores for video frames and leverage the scores as offline supervision to train a keyframe selector. Without subjective and error-prone manual annotations, the proposed framework surpasses previous unsupervised methods in producing high-quality results for generic and instructional video summarization tasks, whose performance even is comparable to those supervised competitors. The code is available at https://github.com/Dixin-s-Lab/Video-Summary-IOT.

CCS CONCEPTS

• **Computing methodologies** → **Video summarization**; **Matching**; **Unsupervised learning**.

KEYWORDS

Video summarization, self-supervised learning, semantic alignment, inverse optimal transport, unbalanced Wasserstein distance

*Corresponding author. Email: dixin.luo@bit.edu.cn

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612087

ACM Reference Format:
Yutong Wang, Hongteng Xu, and Dixin Luo. 2023. Self-supervised Video Summarization Guided by Semantic Inverse Optimal Transport. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3581783.3612087

1 INTRODUCTION

The advancement of multimedia technology leads to an enormous growth of video data. At the same time, retrieving and recommending videos effectively and efficiently becomes challenging, given the massive amount of videos. Video summarization provides a competitive solution to this challenge, representing a long video with informative keyframes. The keyframes compress the video size significantly while retaining its primary content. This technique not only provides significant evidence for video retrieval and recommendation [22, 55] but also helps individuals to preview videos quickly through their keyframes, which has been widely used in many application scenarios, e.g., video editing [15, 31], content filtering [71], and semantic analysis [14, 28].

Many video summarization methods have been proposed in the past few decades, which can be coarsely categorized based on the degree of supervision. Supervised summarization methods train keyframe selectors based on the videos with manually-annotated keyframes [13, 36, 39, 74]. These methods often suffer from insufficient training data because video annotation is time-consuming. Instead of leveraging frame-level annotations, weakly-supervised methods [35, 55] use auxiliary information such as video-level tags [6, 38] and user comments [8, 64] to supervise training processes. While collecting such auxiliary information is much easier than labeling frames, it may not be aligned semantically with the keyframes of videos. As a result, the corresponding weakly-supervised summarization methods are only available in some specific scenarios. Recently, some attempts have been made to achieve unsupervised video summarization [2, 34, 46, 56, 69] without any annotations. Still, the performance of the methods is often inferior to the supervised and weakly-supervised ones due to the lack of semantically-meaningful guidance on the keyframe selection.

In this study, we propose a novel self-supervised framework for video summarization with the help of computational optimal transport (OT) techniques [40]. As depicted in Figure 1, we first leverage a pre-trained caption generator [21] to generate textual descriptions for video shots. The textual descriptions provide initial and coarse semantic guidance for video summarization at the shot level. Then, we map the shots' textual descriptions and the video frames into the same semantic space and align their embeddings. In particular, we optimize a multi-layer perceptron (MLP) to

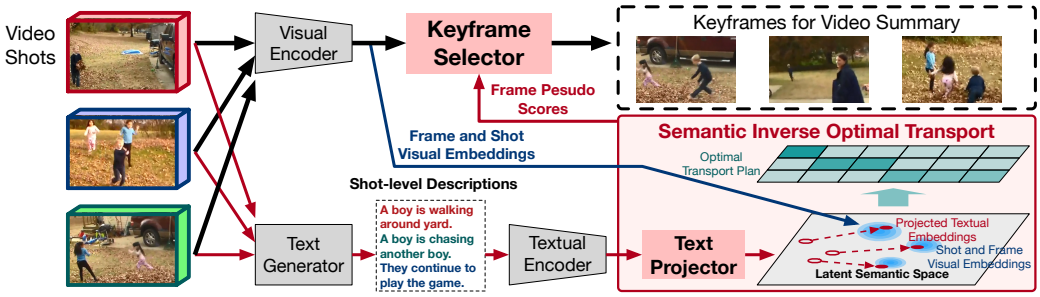

Figure 1: An illustration of our self-supervised video summarization framework. The grey modules are pre-trained models, while the red modules (i.e., the projector and the keyframe selector) are learnable. The red and blue arrows correspond to the steps only used for training, while the black bold arrows are the steps used for both training and testing.

project shot-level textual embeddings to shot-level visual embeddings. At the same time, we compute the unbalanced Wasserstein (UW) distance [42] between the projected textual embeddings and frame-level visual embeddings, in which the MLP determines the underlying distance. Solving these two problems jointly in a bi-level optimization framework leads to an inverse optimal transport (IOT) problem [68]. We solve this problem efficiently by an alternating optimization algorithm, in which a boundary regularizer is applied to avoid trivial solutions. The learned transport plan and underlying distance help to construct frame-level pseudo-significance scores. We train a keyframe selector by taking the scores as the visual embeddings' labels. In the inference phase, we leverage the keyframe selector to predict the scores of video frames and select keyframes by the 0/1 Knapsack algorithm [47].

The proposed self-supervised video summarization framework generates semantic guidance from the videos themselves rather than external or predefined annotations, leading to a novel and promising optimal transport-based solution to unsupervised video summarization. To our knowledge, this framework makes the first attempt to achieve cross-modal semantic alignment based on the inverse optimal transport strategy. The MLP-based projector and the transport plan benefit each other, resulting in more precise alignment results and, thus, more reliable pseudo scores (for offline supervision). We conduct comprehensive experiments on various video summarization tasks, including the generic video summarization task commonly considered by existing methods [34, 36, 39] and the instructional video summarization task proposed in [35]. Experiments show that our approach outperforms existing unsupervised methods in generating high-quality video summaries and can even be comparable to the supervised competitors.

2 RELATED WORK

2.1 Video Summarization

Most existing video summarization methods employ supervised learning strategies [13, 18, 36, 39, 73]. Given frame-level annotations, these methods typically treat the summarization task as a label prediction problem and train various models, e.g., determinantal point process [27], LSTM [30, 73, 74], attention model [3, 13, 18, 66], knowledge-based models [57] and graph-based models [39], to predict keyframes. However, acquiring frame-level annotations from a large number of videos is always challenging because manually labeling frames is with low efficiency and unsatisfactory reliability — such subjective annotations may inherit the bias of annotators. As a result, the supervised video summarization methods often suffer from the data insufficiency issue and poor generalization power.

To reduce the dependency on manual frame-level annotations, weakly-supervised summarization methods are proposed [6, 35, 38, 55] based on the video-level or shot-level labels generated by video platforms or associated with the video themselves, e.g., video-level tags [6, 38], metadata like titles [47] and descriptions [55], and external texts like subtitles and user comments [8]. Based on such labels, many methods apply weakly-supervised algorithms to train their models [35, 55]. Recently, several attempts have been made to accomplish video summarization through unsupervised methods [34, 43, 46, 69]. Typically, the unsupervised summarization methods leverage various adversarial learning frameworks [46, 69], training sequential models to predict keyframes together with discriminators to challenge the quality of the keyframes [34]. However, without fine-grained semantic alignment, these coarse labels introduce significant uncertainty to the selection of keyframes and, thus, harm the performance of the weakly-supervised summarization methods. This problem becomes even worse for unsupervised methods — without any semantically-meaningful guidance, the unsupervised methods have a risk of ignoring important video content and thus fail to produce high-quality video summaries.

2.2 Optimal Transport for Semantic Alignment

Optimal transport (OT) theory [40, 53] provides a series of powerful computational methods for comparing probability distributions, which has been used in domain adaptation [65], image generation [4, 12, 50], word embedding [1, 26, 63], and document comparison [20, 68, 70]. In particular, these OT-based methods provide theoretically-guaranteed solutions to various matching problems, e.g., cross-modal alignment [7, 9, 33], graph matching [62], point cloud registration [5, 41], and densely-packed semantic correspondence issues [32]. For instance, the Vision-Language Pre-trained (VLP) model, Uniter [9], applies optimal transport to promote fine-grained alignment between words and image regions, resulting in better joint embeddings for downstream tasks. The high-performance detector YOLOX [17] utilizes the Optimal Transport Assignment (OTA) [16] as a candidate label-assigning strategy,

given its outstanding matching performance. These applications demonstrate the remarkable versatility of optimal transport in cross-modal alignment and inspire us to develop an OT-based framework for video summarization.

Most existing methods solve optimal transport problems in the Kantorovich form (i.e., Wasserstein distance) [23]. Some efficient approximate algorithms have been proposed, such as Sinkhorn scaling algorithm [10], proximal point method [60], and Bregman alternating direction method of multipliers (B-ADMM) [54, 61]. Recently, inverse optimal transport (IOT) has been proposed to optimize the underlying distance given a noisy or partially observed transport plan [29, 48]. This problem can be solved by the hypergradient method proposed in [59]. The work in [68] applies an IOT-based framework to achieve textual data alignment, which obtains encouraging performance in legal document analysis. In this study, we propose a new IOT-based alignment method for self-supervised video summarization.

3 PROPOSED FRAMEWORK

3.1 Self-supervised Principle

Given an arbitrary video with I frames, denoted as $\mathcal{V} = \{v_i\}_{i=1}^{I}$, where v_i represents the i-th frame, video summarization aims to select a set of keyframes, i.e., $\{v_i\}_{i\in\mathcal{S}}$ and $\mathcal{S} \subset \{1, ..., I\}$, to capture the primary content, especially, the highlights of the video. To achieve this aim, we would like to learn a keyframe selector, denoted as $g : \mathcal{V} \mapsto \{0, 1\}^I$, in a self-supervised learning framework based on a set of unlabeled training videos.

Neither using manually-labeled video frames nor purely relying on visual information of the frames, the proposed self-supervised learning framework leverages a pre-trained multi-modal model as a caption generator, generating textual descriptions from video shots. Then, it extracts useful frame-level information from the generated texts to supervise the training of the keyframe selector. On the one hand, this framework can be more efficient than existing supervised learning methods because the texts are generated automatically rather than manually. On the other hand, compared to weakly-supervised and unsupervised methods, the texts describe the videos themselves and are highly correlated with the visual contents, which can provide semantically-meaningful guidance when training the keyframe selector.

Denote the training videos as $\mathcal{V} = \{(\mathcal{V}_n\}_{n=1}^{N}$. In this study, we first divide each video into several shots with a fixed duration. Applying a pre-trained video caption model [21] as the caption generator, we generate a textual description for each video shot. As a result, we obtain a set of video-text pairs, i.e., $\mathcal{D} = \{(\mathcal{V}_n, \mathcal{W}_n)\}_{n=1}^{N}$. Here, $\mathcal{V}_n = \{v_{i,n}\}_{i=1}^{I_n}$ represents the n-th training video with I_n frames, and $\mathcal{W}_n = \{w_{j,n}\}_{j=1}^{J_n}$ represents J_n sentences corresponding to the J_n shots of the video $\mathcal{V}_n$. Note that, we apply the shot-level textual descriptions because generating texts in the frame level is time-consuming and contain many redundant and repeated sentences, which is unnecessary for our task.

In our self-supervised learning framework, the main challenge is extracting visual and textual features from the above dataset and establishing their semantic correspondence to guide the selection of keyframes. In the following content, we will show that this challenge can be resolved with the help of computational optimal transport techniques. In particular, we apply an inverse optimal transport method to learn an explicit projection from textual embeddings to visual ones and compute the unbalanced Wasserstein between the two domains jointly, leading to informative guidance for training the keyframe selector.

3.2 Semantic Inverse Optimal Transport

For each video-text pair in the dataset, we can extract their features based on the pre-trained multi-modal model. Specifically, given a video with I frame and J shot-level textual descriptions, i.e., $(\mathcal{V} = \{v_i\}_{i=1}^{I}, \mathcal{W} = \{w_j\}_{j=1}^{J})$, we apply the pre-trained CLIP model [45] to obtain D-dimensional visual and textual embeddings, i.e., $\boldsymbol{V} = f_v(\mathcal{V}) = [\boldsymbol{v}_i] \in \mathbb{R}^{I\times D}$ and $\boldsymbol{W} = f_w(\mathcal{W}) = [\boldsymbol{w}_j] \in \mathbb{R}^{J\times D}$, respectively. To build the semantic correspondence between the visual and textual embeddings at the frame level, we need to align them in the latent space. A straightforward way to achieve this alignment task is computing the Wasserstein distance [9, 16] (or its variants [33]) between the visual and textual embeddings, which leads to the optimal transport problem in the Kantorovich form:

$$d_{\mathrm{w}}(\boldsymbol{V}, \boldsymbol{W}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{u},\boldsymbol{\mu})} \mathbb{E}_{(\boldsymbol{v},\boldsymbol{w})\sim\boldsymbol{T}}[d(\boldsymbol{v}, \boldsymbol{w})] = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{u},\boldsymbol{\mu})} \langle \boldsymbol{D}_{vw}, \boldsymbol{T}\rangle \quad (1)$$

Here, $\boldsymbol{D}_{vw} = [d(\boldsymbol{v}_i, \boldsymbol{w}_j)] \in \mathbb{R}^{I\times J}$ represents the distance matrix, where $d(\boldsymbol{v}_i, \boldsymbol{w}_j)$ indicates the Euclidean distance between $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$. $\boldsymbol{u} \in \Delta^{I-1}$ and $\boldsymbol{\mu} \in \Delta^{J-1}$ represent the empirical distributions of visual and textual embeddings, respectively. Without any prior knowledge, we often assume they are uniform, i.e., $\boldsymbol{u} = \frac{1}{I}\mathbf{1}_I$ and $\boldsymbol{\mu} = \frac{1}{J}\mathbf{1}_J$. $\boldsymbol{T} \in \Pi(\boldsymbol{u}, \boldsymbol{\mu}) = \{\boldsymbol{T} \geq 0 | \boldsymbol{T}\mathbf{1}_J = \boldsymbol{u}, \boldsymbol{T}^T\mathbf{1}_I = \boldsymbol{\mu}\}$ is the joint distribution of the visual and textual embeddings, which is also called **transport plan**. As shown in (1), the Wasserstein distance aims to find the optimal transport plan, denoted as $\boldsymbol{T}^*$, such that the expectation of the distance is minimized. The element of $\boldsymbol{T}^*$, i.e., $\boldsymbol{t}^*_{ij}$, represents the coherency probability of $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$, which may indicate the semantic correspondence between the i-th video frame and the j-th textual description.

Applying the above method directly often results in unsatisfactory semantic alignment because of the following two reasons.

(1) Firstly, the rationality of the learned semantic correspondence depends on the quality of the visual and textual embeddings (and accordingly, the underlying distance matrix) [9, 16, 33]. When using the pre-trained CLIP [45], however, we find that the distribution of the visual embeddings is significantly different from that of the textual embeddings. As shown in Figure 2(a), the textual embeddings aggregate together, while the visual embeddings have clustering structures associated with different video shots. These results suggest that the pre-trained CLIP model has poor semantic consistency under the present data distribution.
(2) Secondly, given shot-level textual embeddings and frame-level visual embeddings, we need to achieve a partial alignment, i.e., matching some informative frames with meaningful texts, because many frames, even shots, are redundant or meaningless. In this case, the distributions of the frames and texts are sparse but unknown. In other words, the uniform

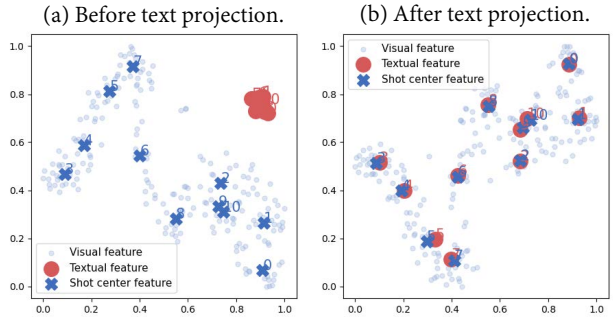

Figure 2: For the "video_13" in SumMe [19], we visualize the t-SNE [51] of the semantic gap between frames and textual descriptions before and after the text projection module. The numbers indicate the order in which the textual description (video shot) appears in the video. The data are from .

marginal distributions and the strict constraints used in (1) are unsuitable for our problem.

We apply a novel inverse optimal transport (IOT) method to solve the above challenges. Given N video-text pairs and the corresponding embeddings $\{(\boldsymbol{V}_n, \boldsymbol{W}_n)\}_{n=1}^N$, we can learn a projector f from the textual domain to the visual domain and a set of optimal transport plans $\{\boldsymbol{T}_n\}_{n=1}^N$ by solving a regularized IOT problem:

$$\begin{aligned} &\underbrace{\min_{\{\boldsymbol{T}_n\}_{n=1}^N, f} \sum_{n=1}^N \langle \boldsymbol{D}_n(f), \boldsymbol{T}_n \rangle + \tau R_T(\boldsymbol{T}_n)}_{\sum_{n=1}^D d_{\text{uw}}(\boldsymbol{V}_n, f(\boldsymbol{W}_n))} \\ &s.t.\ f \in \arg\min_f \underbrace{\sum_{n=1}^N \sum_{j=1}^{J_n} d(\hat{\boldsymbol{v}}_{n,j}, f(\boldsymbol{w}_{n,j})) + R_f(f(\boldsymbol{W}_n))}_{\mathcal{L}_f}. \end{aligned} \tag{2}$$

As shown in (2), the regularized IOT problem corresponds to a bi-level optimization problem. The upper-level problem corresponds to the summation of the unbalanced Wasserstein (UW) distances [42] defined for the N video-text pairs. For each pair, compared to the Wasserstein distance in (1), the main differences in each unbalanced Wasserstein distance include two points and help to overcome the above two challenges. Firstly, we apply the projector f to the textual embeddings before computing the underlying distance matrix, i.e., $\boldsymbol{D}_n(f) = [d(\boldsymbol{v}_{n,i}, f(\boldsymbol{w}_{n,j}))] \in \mathbb{R}^{I_n \times J_n}$. The projector aims to enhance the consistency between the textual domain and the visual domain, such that the underlying distance is more reliable, and accordingly, the transport plan becomes more semantically-meaningful. In this study, we learn the projector and the N transport plans jointly. Secondly, we introduce the regularizer on the marginals of each transport plan, i.e., $R(\boldsymbol{T}_n) = \text{KL}(\boldsymbol{T}_n \boldsymbol{1}_{J_n} \| \frac{1}{I_n} \boldsymbol{1}_{I_n}) + \text{KL}(\boldsymbol{T}_n^\top \boldsymbol{1}_{I_n} \| \frac{1}{J_n} \boldsymbol{1}_{J_n})$, and use a hyperparameter $\tau > 0$ to control its significance. This regularizer penalizes the KL-divergence between the marginals of the transport matrix and the corresponding uniform distributions. The unbalanced Wasserstein distance relaxes the strict marginal constraints of the transport plan to the regularizer. It thus allows us to perform partial alignment — only a part of the frames are aligned to the texts, which helps to select the informative frames and filter out the redundant ones.

Because learning the projector and the transport plans jointly may lead to trivial solutions, we introduce a lower-level optimization problem to regularize the learning of the projector. Specifically, the correspondence between each video's textual descriptions and shots is known because we generate one textual description per shot. Therefore, in the lower-level optimization problem, we consider learning the projector to minimize the distance from each projected textual embedding to the visual embedding of its corresponding video shot. Here, we take the average of the frames' visual embeddings within a shot as the shot's visual embedding, denoted as $\widehat{\boldsymbol{V}} = [\hat{\boldsymbol{v}}_j] \in \mathbb{R}^{J \times D}$. Additionally, when dividing continuous video content into fixed-length video shots, we may introduce some undesired semantic ambiguity at the boundaries of those adjacent shots, i.e., the text embeddings corresponding to adjacent shots appear close to each other in the latent space. To avoid a projector mapping different textual embeddings to the same point), for each video, we consider a boundary regularizer based on the KL-divergence:

$$\mathcal{R}_f(f(\boldsymbol{W}_n)) := \text{KL}(\boldsymbol{D}_w(f(\boldsymbol{W}_n)) \| \boldsymbol{B}_n). \tag{3}$$

Here, $\boldsymbol{D}_w(f(\boldsymbol{W})) = [d(f(\boldsymbol{w}_{n,j}), f(\boldsymbol{w}_{n,j'}))] \in \mathbb{R}^{J_n \times J_n}$ is the Euclidean distance matrix for the projected textual embeddings. $\boldsymbol{B}_n = \boldsymbol{1}_{J_n \times J_n} - \boldsymbol{I}_{J_n}$ is a boundary constraint matrix of size $J \times J$, whose diagonal elements are zeros and the remaining elements are ones. The KL divergence penalizes the difference between $\boldsymbol{D}_w(f(\boldsymbol{W}))$ and $\boldsymbol{B}_n$, encouraging the diversity of the projected textual embeddings.

We apply an alternating optimization strategy to solve the IOT problem. Specifically, given the current projector, we solve N unbalanced Wasserstein distance problems based on the Bregman ADMM algorithm in [33, 61] and obtain N optimal transport plans accordingly. Then, we update the projector by Adam [25], in which the objective function considers both the lower-level and the upper-level problems, and the gradients of f with respect to the optimal transport plans are computed based on the hypergradient method in [59]. In this study, the architecture of the projector f is a Multi-Layer Perceptron (MLP) that contains two fully-connected (FC) layers with a ReLU layer between them. As shown in Figure 2(b), applying the learned f, the projected textual embeddings aligned well with the shots' visual embeddings. Based on the well-aligned projected textual embeddings and the frame-level visual embeddings, we can obtain reliable transport plans to guide the learning of the keyframe selector.

3.3 Pseudo Scores for Keyframe Selection

For each video, we construct the frame-level pseudo-significance score to supervise the training of the keyframe selector. In particular, the proposed pseudo-significance score consists of the following two parts: the alignment score and the representation score.

Alignment score. The alignment score is based on the fact that one textual description can capture the key information of a given video shot. Thus, the alignment between the text and the frames within the shot can reveal the significance of each frame relative to the shot. After solving the IOT problem, the optimal transport plan we learned has achieved semantic alignment. Given a video with I frames and J textual descriptions, we denote the corresponding optimal transport plan as $\boldsymbol{T}^* = [t_{ij}^*] \in \mathbb{R}^{I \times J}$. For the frame v_i, we can select the top K texts that match the frame best

Algorithm 1 Training steps of our method

Require: $\mathcal{V} = \{\mathcal{V}_n\}_{n=1}^N$; learning rates η_1, η_2; trade-off weight γ.
Generate texts $\mathcal{W} = \{\mathcal{W}_n\}_{n=1}^N$ based on $\mathcal{V} = \{\mathcal{V}_n\}_{n=1}^N$.
Extract embeddings $\mathcal{D} = \{(\boldsymbol{V}_n, \boldsymbol{W}_n)\}_{n=1}^N$ via a pre-trained representation model.
▷ **Solving the IOT problem**
repeat
  Sample $(\boldsymbol{V}_n, \boldsymbol{W}_n)$ from $\mathcal{D}$. Calculate shot embeddings $\hat{\boldsymbol{V}}_n$.
  Calculate the lower-level objective $\mathcal{L}_f$.
  Calculate the UW distance $d_{\text{uw}}(\boldsymbol{V}_n, f(\boldsymbol{W}_n))$.
  Calculate $\mathcal{L} = \mathcal{L}_f + \gamma d_{\text{uw}}$, then $\theta_f \leftarrow \theta_f - \eta_1 \nabla_{\theta_f} \mathcal{L}$.
until convergence
▷ **Pseudo score generation**
for $n = 1, ..., N$ **do**
  Compute alignment score $\boldsymbol{s}_{n,a}$ via $\boldsymbol{T}^*$ of $d_{\text{uw}}(\boldsymbol{V}_n, f(\boldsymbol{W}_n))$.
  Compute representation score $\boldsymbol{s}_{n,r}$ via (4).
  Compute pseudo score s_n via (5).
end for
▷ **Training keyframe selector**
repeat
  Sample $\{\boldsymbol{V}_n, \boldsymbol{W}_n, s_n\}$ randomly from $\mathcal{D} \cup \{s_n\}_{n=1}^N$.
  Calculate importance scores $\boldsymbol{y}_n$ by $g(\boldsymbol{V}_n)$.
  Calculate $\mathcal{L}_g$ via (6), then $\theta_g \leftarrow \theta_g - \eta_2 \nabla_{\theta_g} \mathcal{L}_g$.
until convergence
Select keyframes by 0/1 Knapsack algorithm based on $\{\boldsymbol{y}_n\}_{n=1}^N$.

by $\mathcal{J}_i = \arg\text{sort-}K_{j\in\{1,...,J\}} t_{ij}$. Similarly, for the text w_j, we can select the top K frames by $\mathcal{I}_j = \arg\text{sort-}K_{i\in\{1,...,I\}} t_{ij}$. Accordingly, denote the alignment score as $\boldsymbol{s}_a = [s_{a,i}] \in \mathbb{R}^I$. We consider the following two implementations in this study:

(1) **Frame-oriented score:** Given the text set $\mathcal{J}_i$, we define the i-th frame's alignment score as $s_{a,i} = \sum_{j\in\mathcal{J}_i} -d(\boldsymbol{v}_i, f(\boldsymbol{w}_j))$.
(2) **Text-oriented score:** Given the set of the top K frames $\mathcal{I}_j$, for the j-th text, we define the relative significance of the i-th frame as $s_{ij} = \frac{\sum_{k\in\mathcal{I}_j} d(\boldsymbol{v}_k, f(\boldsymbol{w}_j))}{d(\boldsymbol{v}_i, f(\boldsymbol{w}_j))}$. Finally, the alignment score of the i-th frame is set as $s_{a,i} = \max_{j\in\{1,...,J\}} s_{ij}$.

Both implementations ensure that the frames matching the textual descriptions have high alignment scores.

Representation score. Besides the alignment score, we further propose a representation score for video frames, denoted as $\boldsymbol{s}_r = [s_{r,i}] \in \mathbb{R}^I$, which measures the frames' representativeness. This score is motivated by the requirement of video summarization on the completeness of the original video's storyline — the selected keyframes should be able to reconstruct the frames in their neighborhoods with tolerable errors. We define the representation score based on the reconstruction error defined in the latent space. For the i-th frame, we consider its K' neighbor frames, denoted as a set $\mathcal{I}_i$. The i-th representation score is defined as

$$s_{r,i} = \frac{1}{|\mathcal{I}_i|} \sum\nolimits_{k\in\mathcal{I}_i} \|\boldsymbol{v}_i - \boldsymbol{v}_k\|_2, \tag{4}$$

which calculates the average of the Euclidean distances between the frame's embedding and its neighbors' embeddings. The lower the representation score is, the more representative the frame is.

Considering the above two kinds of scores, we obtain the proposed frame-level pseudo-significant score as follows.

$$\boldsymbol{s} = 0.5(\hat{\boldsymbol{s}}_a - \hat{\boldsymbol{s}}_r + 1), \tag{5}$$

where $\hat{\boldsymbol{s}}_a$ and $\hat{\boldsymbol{s}}_r$ are normalized alignment and representation scores.[1] The pseudo-significance score provides us with "labels" to the proposed keyframe selector g, which comprises an encoder-only transformer with positional embeddings [52]. Given N training videos, whose frame-level visual embeddings are $\{\boldsymbol{V}_n \in \mathbb{R}^{I_n\times D}\}_{n=1}^N$ and pseudo-significant score vectors are $\{\boldsymbol{s}_n\}_{n=1}^N$, we learn the keyframe selector as follows:

$$\min_g \sum\nolimits_{n=1}^N \sum\nolimits_{i=1}^{I_n} \text{loss}(g(\boldsymbol{V}_n), \boldsymbol{s}_n), \tag{6}$$

where the loss function can be the mean-square-error (MSE) or the binary cross-entropy loss. The training scheme of our self-supervised video summarization method is shown in Algorithm 1.

3.4 Keyframe Inference

Following prior methods [13, 36, 46, 73], we achieve keyframe selection in the inference phase based on the 0/1 Knapsack algorithm [47]. In particular, given a video with I frames , we first extract I visual embeddings based on the pre-trained model and compute their significance scores based on the learned keyframe selector, i.e., $\boldsymbol{y} = g_{\text{kfs}}(f_v(\mathcal{V}))$. Then we select at most L keyframes by solving the following binary optimization problem:

$$\boldsymbol{u}^* = \arg\max\nolimits_{\boldsymbol{u}\in\{0,1\}^I} \langle \boldsymbol{u}, \boldsymbol{y}\rangle \quad \text{s.t. } \|\boldsymbol{u}\|_1 \le L. \tag{7}$$

The objective function in (7) corresponds to maximizing the scores of the selected frames, and the constraint ensures that the maximum number of the selected frames is L. This is a typical Knapsack problem and can be solved by the 0/1 Knapsack algorithm [47]. As a result, we take the selected keyframes as the video summary.

4 EXPERIMENTS

4.1 Implementation Details

To demonstrate the effectiveness of our video summarization approach, we test it on both generic and instructional video summarization tasks and compare it with state-of-the-art methods.

4.1.1 Tasks and Datasets. The video summarization task generally refers to the **generic video summarization** [44, 46, 69, 72], which aims to provide a comprehensive summary of the original video by selecting important shots/frames that cover the main content. Recently, a new type of video summarization task called **instructional video summarization** [35] has been proposed, which aims to summarize an instructional video by the shots/frames corresponding to the steps of a specific task shown in the video. In the generic video summarization task, we consider **SumMe** [19] and **TVSum** [47] datasets. SumMe comprises 25 user videos ranging from 1.5 to 6.5 minutes, annotated by mostly 15 users. TVSum consists of 50 videos ranging from 1 to 11 minutes, annotated by 20 users. For each dataset, we follow the data splits in [13], using 80% data for training and 20% data for testing. Besides the above *standard data setting*, we augment the training data by two more external video

[1] In this study, for each score vector, we minimize its minimum and divide its dynamic range, such that the range of the normalized score is $[0, 1]$. Accordingly, the range of the final pseudo score is $[0, 1]$ as well.

Category	Method	SumMe			TVSum			WikiHow		
		Standard	Augment	Transfer	Standard	Augment	Transfer	F1-Score	Precision	Recall
Supervised	SUM-GAN [34]	38.3	43.1	41.4	55.9	58.3	56.6	42.3	48.5	46.2
	*RSGN [75]	45.0	45.7	44.0	60.1	61.1	60.0	-	-	-
	DSNet [77]	48.6	49.1	46.3	61.9	63.3	59.4	49.4	54.5	59.5
	VASNet [13]	50.0	51.4	44.2	61.4	62.3	57.2	49.6	56.0	55.9
	*SumGraph [39]	51.4	52.9	48.7	63.9	65.8	60.5	-	-	-
Unsupervised or Self-supervised	SUM-GAN [34]	36.7	39.3	41.8	54.5	57.6	56.6	46.3	52.3	52.1
	DR-DSN [76]	37.3	42.5	41.6	55.8	58.4	57.4	48.0	53.0	53.2
	*RSGN [75]	42.3	43.6	41.2	58.0	59.1	**59.7**	-	-	-
	IV-Sum [35]	-	-	-	-	-	-	43.6	50.8	50.6
	CLIP-It [36]	**50.8**	48.3	43.8	57.9	**60.5**	58.9	49.6	**56.9**	53.1
	Ours	50.2	**48.8**	**44.9**	**59.4**	60.3	59.2	**55.2**	53.4	**70.8**

Table 1: Comparisons for various video summarization methods. For SumMe and TVSum, we only list the F1-Scores achieved under the "Standard", "Augment", and "Transfer" settings. For WikiHow, we list the F1-Score, Precision, and Recall achieved under the Standard data setting. The methods with "*" do not release source codes, so we quote their results from the references. For the unsupervised and self-supervised methods, we bold the best results and underline the second best ones.

datasets, i.e., Open Video Project (OVP) and Youtube [11],[2] leading to an *augmented data setting*. Furthermore, we consider a *transfer learning setting*, using TVSum (SumMe), OVP, and Youtube for training and SumMe (TVSum) for testing. In the instructional video summarization task, we apply the **WikiHow** dataset [35], which contains 2,105 videos of 20 various tasks. We partition WikiHow into 1,684 training videos and 421 testing videos.

4.1.2 Baselines. In both generic and instructional video summarization tasks, we compare our approach with state-of-the-art methods. Our comparison primarily focuses on existing unsupervised or self-supervised methods, including SUM-GAN [34], DR-DSN [76], RSGN [75], SumGraph [39], CLIP-It [36], and IV-Sum [35].[3] Additionally, we conduct comparative experiments with some supervised methods such as DSNet [77], VASNet [13], and the supervised version of SUM-GAN [34], and RSGN [75].

4.1.3 Evaluation Metrics. Following the work in [35–37], we use F1-Score (and associated Precision and Recall) to evaluate each method. Given a generated summary and the corresponding ground truth summary, each represented as a list of keyframes' indices, we locate the keyframes of each summary in the original video and represent them as two binary sequences with the same length, denoted as A and B, respectively. We compute the precision and recall as $P = |A \cap B| / |A|$ and $R = |A \cap B| / |B|$, respectively. The F1-Score is $F1 = \frac{2PR}{P+R}$. Given the frame-level scores associated with A and B, we further apply the correlation coefficients (i.e., Kendall's τ [24] and Spearman's ρ [78]) to evaluate each method. Following the strategy in [35–37], for each video with M manual annotations, we compute each evaluation metrics M times independently based on the annotations and take the average value as the final result for TVSum while the maximum one for SumMe.

4.1.4 Model Configuration. **Text generation.** Given an input video, we divide the video into multiple shots (e.g., 10s per shot) and take each shot as the input of a captioning model. For our framework and IV-Sum, we apply the Hierarchical Temporal-Aware (HiTeA) Video-Language Pre-training model [67] to generate one sentence per shot. For CLIP-It, we follow its default setting, applying the Bi-Modal Transformer (BMT) model [21] to generate textual descriptions. The BMT model inputs a shot and outputs multiple sentences corresponding different durations. We choose the sentence with the longest duration as the textual description.

Image and text encoding. Following [44, 46, 69, 72], we downsample the video frames uniformly at two frames per second to reduce temporal redundancy and computational demand. By default, both the frames and the above shot-level textual descriptions are encoded by a pre-trained CLIP [45], and the dimension of the visual and textual embeddings is 512. For the ablation study, we further conduct experiments using GoogleNet [49] for image encoding and S3D [58] for both image and text encoding.

Model and hyperparameter settings. We implement the textual projector f as a two-layer MLP and the keyframe selector g as a Transformer with eight heads and six encoding layers [52]. When training f, we apply the Adam [25] with a learning rate of 10^{-4} and 150 epochs. When training g, we apply the Adam with a learning rate of 10^{-5} and 50 epochs, and the text-oriented alignment scores are used. More details of our method are given in Appendix.

4.2 Quantitative and Qualitative Comparisons

Table 1 presents the experimental results of various video summarization methods on the three datasets. Among the unsupervised and self-supervised methods, our method achieves the top-2 performance consistently in all situations. Especially in the instructional video summarization task, our method significantly surpasses all supervised and unsupervised baselines in F1-Score and Recall. In particular, the precision of our method is comparable to the baselines. At the same time, its recall is much higher, which means that the summaries generated by our method can cover the manual annotations with high accuracy. These results implicitly support our claim that self-supervised learning is a promising solution to instructional video summarization.

[2]The OVP contains 50 videos with varying lengths between 1 to 4 minutes, and the Youtube dataset contains 39 videos ranging from 1 to 10 minutes.
[3]IV-Sum [35] is a method specialized for instructional video summarization.

Datasets	SumMe		TVSum	
Metrics	τ	ρ	τ	ρ
Human (Oracle)	0.205	0.213	0.177	0.204
DR-DSN [76]	0.024	0.029	-0.016	-0.019
SUM-GAN [34]	0.035	0.043	0.017	0.020
CLIP-It [36]	0.046	0.051	0.038	0.041
Ours	**0.069**	**0.056**	**0.097**	**0.079**

Table 2: Comparing Kendall's τ and Spearman's ρ for generic video summarization on the SumMe and TVSum datasets.

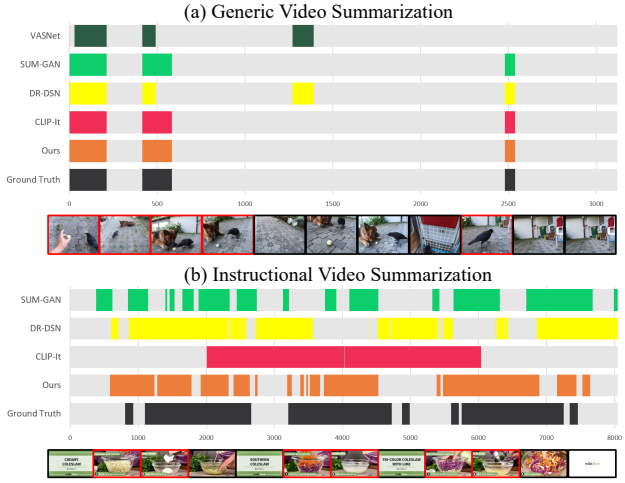

Figure 3: Qualitative comparison for the ground-truth summary and the summaries achieved by SUM-GAN [34], DR-DSN [76], VASNet [13], CLIP-It [36] and our method.

Our method outperforms the unsupervised and self-supervised baselines for the generic video summarization task in most situations. Especially in the challenging "Transfer" setting, our method achieves significant improvements compared to most baselines, demonstrating our method's generalization power. Additionally, we compare our method with the baselines on the correlation coefficients achieved in the generic video summarization task. The results in Table 2 further verify the superiority of our method to its competitors. Moreover, we compute the averaged correlation coefficient between the different manual annotations of the same video (i.e., the "Human" row in Table 2). We can find that the correlation between different manual annotations is much higher than that between the predicted and manual annotations, which means that taking one human's summary as a reference, the summaries generated by existing learning-based methods are still worse than humans' summaries on semantic consistency. Compared to existing methods, our method significantly progresses in reducing the gap.

We present an evaluation of our proposed method and the baseline approaches through user studies. For the five test videos in split0 from the SumMe dataset, we generated video summaries using our method, SUM-GAN [34], DR-DSN [76], and CLIP-It [36]. Five volunteers were invited to assess and vote on the completeness of the summary content, camera switching, and other relevant factors for each video's corresponding four summaries. A total of 25 votes were collected. The summarized vote counts for the four methods are presented in Table 3. Notably, our method received a higher number of votes compared to the other baselines. Furthermore, it is worth mentioning that while CLIP-It had a higher objective score than DR-DSN, the subjective score of DR-DSN was rated higher than CLIP-It.

	SUM-GAN	CLIP-It	DR-DSN	Ours
Votes	2	5	6	12

Table 3: Comparisons on votes for various methods on test videos of SumMe dataset.

Scoring Method	SumMe (F1)	TVSum (F1)	WikiHow (F1)
Frame-oriented $\hat{s}_a$	46.0	58.1	54.0
Text-oriented $\hat{s}_a$	48.9	58.5	54.8
$1 - \hat{s}_r$	45.2	59.0	54.2
Proposed	**50.2**	**59.4**	**55.2**

Table 4: Ablation study of scoring methods.

f_w	f_v	F1-Score	Precision	Recall
CLIP	GoogleNet	43.9	43.0	46.5
S3D	S3D	44.5	43.5	46.0
CLIP	CLIP	50.2	49.5	51.5

Table 5: Ablation study of pre-trained embedding models.

Figure 3 provides some representative summarization results. In particular, Figure 3(a) visualizes the summaries generated by various methods for the "video_25" in SumMe. This video portrays a man engaging in a passing game with a dog and a bird. The ground-truth summary shows the interactions among people, the dog, and the bird. Our method and CLIP-It successfully capture the highlights in the video. On the contrary, other methods either miss important highlights or inaccurately capture non-key frames (e.g., blank backgrounds and uneventful scenes). Figure 3(2) shows the summaries for the video "Make-Coleslaw" in WikiHow. This video presents three methods to make coleslaw, each containing four steps. Our method outperforms its competitors on the completeness and accuracy of the summary.

4.3 Ablation Study

4.3.1 Rationality of Scoring Method. In the above experiments, we generate frame-level pseudo-significance scores by (5), in which the text-oriented alignment scores are applied. To demonstrate the rationality of this scoring method, we consider three more settings of the pseudo scores: *i*) normalized frame-oriented alignment scores, *ii*) normalized text-oriented alignment scores, and *iii*) one minus normalized representation scores. We learn the keyframe selector for each setting and report the summarization results in Table 4. Firstly, we can find that text-oriented alignment scores yield better performance than frame-oriented alignment scores. One reason for this phenomenon is that the text-oriented scoring method ensures

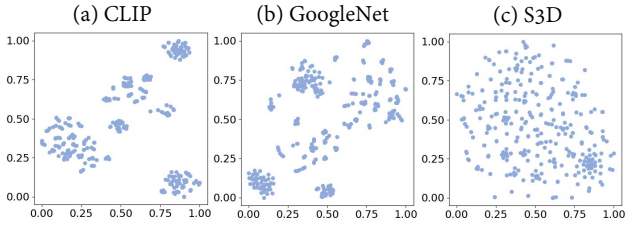

Figure 4: The t-SNE plots of CLIP, GoogleNet and S3D visual embeddings for the "video_14" in TVSum.

each text must be assigned to some frames with a high score. In contrast, the frame-oriented scoring method can possibly miss some textual descriptions. Secondly, the experimental results show that jointly considering the alignment and representation scores can improve our method's performance.

4.3.2 Impact of Embedding Model. The pre-trained embedding models impact the performance of our method. In the above experiments, we use CLIP to encode images and texts. In Table 5, we consider other settings and observe the following phenomena.

Firstly, when replacing CLIP with GoogleNet for the visual modality, the performance of our method degrades. This is because the CLIP for textual embedding and the GoogleNet for visual embedding are independent pre-training models whose latent spaces are unaligned. Using independent visual and textual embedding models makes semantic alignment harder. As a result, it is more difficult for us to learn a satisfactory textual projector.

Secondly, the embeddings obtained by S3D are inferior to those of CLIP, leading to performance degradation. Figure 4 shows the t-SNE plots of the embeddings obtained by different methods. We can find that the clustering structure of the CLIP embeddings is apparent, with each cluster representing a video shot. The GoogleNet embeddings also have an obvious clustering structure, but they have different distributions compared to the CLIP embeddings and thus make the learning task harder, as aforementioned. The S3D embeddings lack evident clustering characteristics and thus provide little semantic guidance for video summarization.

4.3.3 Impact of Text Generator. Two captioning models, including the Bi-Modal Transformer (BMT) model [21] and the Hierarchical Temporal-Aware Video-Language Pre-training framework (HiTeA) in [67], are considered as text generator. In terms of text characteristics, the texts generated by the BMT model contain more nouns and verbs, and the textual descriptions are more detailed, but the accuracy of the texts is relatively low and sometimes there are grammatical errors. In contrast, the texts generated by HiTeA are shorter but more accurate. Table 6 shows the performance by using the texts generated by the BMT model and HiTeA, respectively. Observing this table, we can infer that our model is robust to different text types and performs well on both types.

4.3.4 Impact of Model Architecture. The architecture of the proposed keyframe selector significantly impacts the model performance. Besides the default Transformer-based model, we implement the keyframe selector by a Bi-directional LSTM (BiLSTM). The BiLSTM architecture includes one linear layer and two LSTM layers,

Text generator	F1-Score	Precision	Recall
BMT [21]	59.3	59.3	59.3
HiTeA [67]	59.4	59.5	59.4

Table 6: Ablation study of text generators.

Keyframe selector g	SumMe (F1)	TVSum (F1)	WikiHow (F1)
Bi-LSTM	46.3	58.3	55.0
Transformer	50.2	59.4	55.2

Table 7: Ablation study of architectures of keyframe selector.

accommodating bi-directional characteristics. Table 7 shows the performance of the Transformer and the BiLSTM, respectively. We can find that the Transformer-based keyframe selector works better than the BiLSTM model. In our opinion, although both architectures establish temporal dependencies between video frames, the Transformer architecture may be more appropriate for video summarization tasks because its attention mechanism assigns learnable weights to frames explicitly, which matches well with the goal of keyframe selection.

5 CONCLUSION

We have proposed a novel self-supervised framework for video summarization based on computational optimal transport techniques. In particular, we generate textual descriptions for video shots based on a pre-trained cross-modal generative model and align the textual descriptions with video frames by solving an inverse optimal transport problem. The alignment results help to design frame-level pseudo-significance scores, which provide informative supervision for learning the keyframe selector. Experimental results demonstrate that our self-supervised method outperforms state-of-the-art unsupervised and self-supervised baselines and even yields comparable results to some supervised methods. In the future, we plan to collaborate with some media platforms and test our summarization method in real-world scenarios. Additionally, we are interested in trying video foundation models and large language models for caption generation. We would also like to develop a more efficient algorithm to solve the IOT problem.

ACKNOWLEDGMENTS

Dr. Dixin Luo was supported by the National Natural Science Foundation of China (62102031) and the Young Scholar Program (XSQD-202107001) from Beijing Institute of Technology. Dr. Hongteng Xu was supported by the National Natural Science Foundation of China (62106271, 92270110), CAAI-Huawei MindSpore Open Fund, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. He thanks the support from the Beijing Key Laboratory of Big Data Management and Analysis Methods, the Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative. The authors thank the volunteers for supporting the subjective experiments.

REFERENCES

[1] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Gromov-Wasserstein Alignment of Word Embedding Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 1881–1890. https://doi.org/10.18653/v1/d18-1214

[2] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. 2020. AC-SUM-GAN: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 8 (2020), 3278–3292.

[3] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. 2022. Explaining video summarization based on the focus of attention. In *2022 IEEE International Symposium on Multimedia (ISM)*. IEEE, 146–150.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[5] Nicolas Bonneel and David Coeurjolly. 2019. Spot: sliced partial optimal transport. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.

[6] Sijia Cai, Wangmeng Zuo, Larry S Davis, and Lei Zhang. 2018. Weakly-supervised video summarization using variational encoder-decoder and web prior. In *Proceedings of the European conference on computer vision (ECCV)*. 184–200.

[7] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. 2022. Prompt Learning with Optimal Transport for Vision-Language Models. *arXiv preprint arXiv:2210.01253* (2022).

[8] Xu Chen, Yongfeng Zhang, Qingyao Ai, Hongteng Xu, Junchi Yan, and Zheng Qin. 2017. Personalized key frame recommendation. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 315–324.

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*. Springer, 104–120.

[10] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[11] Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern recognition letters* 32, 1 (2011), 56–68.

[12] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. 2018. Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3483–3491.

[13] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. 2018. Summarizing videos with attention. In *Asian Conference on Computer Vision*. Springer, 39–54.

[14] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. 2018. Challenges on large scale surveillance video analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 69–76.

[15] Tsu-Jui Fu, Xin Eric Wang, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. 2022. M3L: Language-based video editing via multi-modal multi-level transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10513–10522.

[16] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. 2021. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 303–312.

[17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430* (2021).

[18] Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. 2021. Supervised video summarization via multiple feature sets with parallel attention. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6s.

[19] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*. Springer, 505–520.

[20] Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Fei Sha, and Kilian Q. Weinberger. 2016. Supervised Word Mover's Distance. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 4862–4870. https://proceedings.neurips.cc/paper/2016/hash/10c66082c124f8afe3df4886f5e516e0-Abstract.html

[21] Vladimir Iashin and Esa Rahtu. 2020. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. *arXiv preprint arXiv:2005.08271* (2020).

[22] Atsushi Kanehira, Luc Van Gool, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Aware video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7435–7444.

[23] L Kantorovich. 1942. On the transfer of masses (in Russian). In *Doklady Akademii Nauk*, Vol. 37. 227.

[24] Maurice G Kendall. 1945. The treatment of ties in ranking problems. *Biometrika* 33, 3 (1945), 239–251.

[25] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[26] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=H196sainb

[27] Frédéric Lavancier, Jesper Møller, and Ege Rubak. 2015. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 77, 4 (2015), 853–877.

[28] Deborah Levy, Yuval Belfer, Elad Osherov, Eyal Bigal, Aviad P Scheinin, Hagai Nativ, Dan Tchernov, and Tali Treibitz. 2018. Automated analysis of marine video with limited data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1385–1393.

[29] Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. 2019. Learning to match via inverse optimal transport. *Journal of machine learning research* 20 (2019).

[30] Yandong Li, Liqiang Wang, Tianbao Yang, and Boqing Gong. 2018. How local is the local diversity? reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 151–167.

[31] Qin Lin, Nuo Pang, and Zhiying Hong. 2021. Automated multi-modal video editing for ads video. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4823–4827.

[32] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. 2020. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4463–4472.

[33] Dixin Luo, Yutong Wang, Angxiao Yue, and Hongteng Xu. 2022. Weakly-Supervised Temporal Action Alignment Driven by Unbalanced Spectral Fused Gromov-Wasserstein Distance. In *Proceedings of the 30th ACM International Conference on Multimedia*. 728–739.

[34] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 202–211.

[35] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. 2022. TL; DW? Summarizing Instructional Videos with Task Relevance & Cross-Modal Saliency. *arXiv preprint arXiv:2208.06773* (2022).

[36] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. 2021. CLIP-It! language-guided video summarization. *Advances in Neural Information Processing Systems* 34 (2021), 13988–14000.

[37] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. 2019. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7596–7604.

[38] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K Roy-Chowdhury. 2017. Weakly supervised summarization of web videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 3657–3666.

[39] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. 2020. Sumgraph: Video summarization via recursive graph modeling. In *European Conference on Computer Vision*. Springer, 647–663.

[40] Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 5-6 (2019), 355–607.

[41] Gabriel Peyré, Marco Cuturi, and Justin Solomon. 2016. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*. PMLR, 2664–2672.

[42] Khiem Pham, Khang Le, Nhat Ho, Tung Pham, and Hung Bui. 2020. On unbalanced optimal transport: An analysis of sinkhorn algorithm. In *International Conference on Machine Learning*. PMLR, 7673–7682.

[43] Aniwat Phaphuangwittayakul, Yi Guo, Fangli Ying, Wentian Xu, and Zheng Zheng. 2021. Self-attention recurrent summarization network with reinforcement learning for video summarization task. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[44] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-specific video summarization. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer, 540–555.

[45] 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 *International conference on machine learning*. PMLR, 8748–8763.

[46] Mrigank Rochan and Yang Wang. 2019. Video summarization by learning from unpaired data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7902–7911.

[47] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition. 5179–5187.

[48] Andrew M Stuart and Marie-Therese Wolfram. 2020. Inverse optimal transport. *SIAM J. Appl. Math.* 80, 1 (2020), 599–619.

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[50] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558* (2017).

[51] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[53] Cédric Villani. 2009. *Optimal transport: old and new*. Vol. 338. Springer.

[54] Huahua Wang and Arindam Banerjee. 2014. Bregman alternating direction method of multipliers. *Advances in Neural Information Processing Systems* 27 (2014).

[55] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang, and Chen Yao. 2018. Video summarization via semantic attended networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[56] Guande Wu, Jianzhe Lin, and Cláudio T Silva. 2021. Era: Entity relationship aware video summarization with wasserstein gan. *arXiv preprint arXiv:2109.02625* (2021).

[57] Jiehang Xie, Xuanbai Chen, Shao-Ping Lu, and Yulu Yang. 2022. A Knowledge Augmented and Multimodal-Based Framework for Video Summarization. In *Proceedings of the 30th ACM International Conference on Multimedia*. 740–749.

[58] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*. 305–321.

[59] Yujia Xie, Yixiu Mao, Simiao Zuo, Hongteng Xu, Xiaojing Ye, Tuo Zhao, and Hongyuan Zha. 2021. A Hypergradient Approach to Robust Regression without Correspondence. In *International Conference on Learning Representations*.

[60] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2020. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*. PMLR, 433–453.

[61] Hongteng Xu. 2020. Gromov-Wasserstein factorization models for graph clustering. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (2020), 6478–6485.

[62] Hongteng Xu, Dixin Luo, and Lawrence Carin. 2019. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems* 32 (2019).

[63] Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. 2018. Distilled Wasserstein Learning for Word Embedding and Topic Modeling. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 1723–1732. https://proceedings.neurips.cc/paper/2018/hash/22fb0cee7e1f3bde58293de743871417-Abstract.html

[64] Linli Xu and Chao Zhang. 2017. Bridging video content and comments: Synchronized video description with temporal summarization of crowdsourced time-sync comments. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[65] Renjun Xu, Pelen Liu, Yin Zhang, Fang Cai, Jindong Wang, Shuoying Liang, Heting Ying, and Jianwei Yin. 2021. Joint partial optimal transport for open set domain adaptation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2540–2546.

[66] Ming Yao, Yu Bai, Wei Du, Xuejun Zhang, Heng Quan, Fuli Cai, and Hongwei Kang. 2022. Multi-Level Spatiotemporal Network for Video Summarization. In *Proceedings of the 30th ACM International Conference on Multimedia*. 790–798.

[67] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. 2022. HiTeA: Hierarchical Temporal-Aware Video-Language Pre-training. *arXiv preprint arXiv:2212.14546* (2022).

[68] Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable legal case matching via inverse optimal transport-based rationale extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 657–668.

[69] Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. 2019. Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9143–9150.

[70] Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin M. Solomon. 2019. Hierarchical Optimal Transport for Document Representation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 1599–1609. https://proceedings.neurips.cc/paper/2019/hash/8b5040a8a5baf3e0e67386c2e3a9b903-Abstract.html

[71] Chen Zhang, Qiang Cao, Hong Jiang, Wenhui Zhang, Jingjun Li, and Jie Yao. 2020. A fast filtering mechanism to improve efficiency of large-scale video analytics. *IEEE Trans. Comput.* 69, 6 (2020), 914–928.

[72] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1059–1067.

[73] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European conference on computer vision*. Springer, 766–782.

[74] Ke Zhang, Kristen Grauman, and Fei Sha. 2018. Retrospective encoders for video summarization. In *Proceedings of the European conference on computer vision (ECCV)*. 383–399.

[75] Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. 2021. Reconstructive sequence-graph network for video summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 5 (2021), 2793–2801.

[76] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[77] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. 2020. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing* 30 (2020), 948–962.

[78] Daniel Zwillinger and Stephen Kokoska. 1999. *CRC standard probability and statistics tables and formulae*. Crc Press.

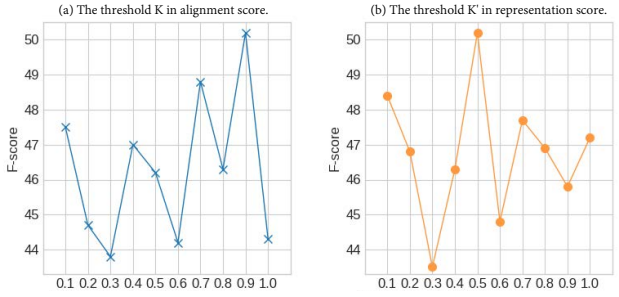

Figure 5: The influence of K and K' in alignment score and representation score, respectively, on SumMe dataset.

Method	Standard	Augment	Transfer
AC-SUM-GAN [2]	47.1	50.0	43.6
DSR-RL [43]	49.1	36.1	35.5
MSVA [18]	47.7	-	-
Ours	50.2	48.8	44.9

Table 8: Quantitative results of F1-Score for generic video summarization task on SumMe dataset.

A IMPLEMENTATION DETAILS

A.1 Bregman ADMM Algorithm

We calculate the unbalanced Wasserstein distance in (2) by applying the Bregman alternating direction method of multipliers (B-ADMM) [33, 54, 61]. More detailed information can be obtained in US-FGW[33].

In this study, the weight of the unbalanced regularizer is 0.5, and the B-ADMM parameters are set to the maximum of 2000 iterations, error bound of 1e-2, and ρ of 1e-2.

A.2 The Key Shot Inference

In the experiments, some tasks require us to summarize videos in the shot level. Therefore, we need to extend our keyframe inference method for key shot inference. Specifically, in the testing phase, given a video $\mathcal{V}$ with I frames, we detect scene change points by the Kernel Temporal Segmentation (KTS) method [44], which divides the input video into M video shots. Given the frame-level significance scores derived by our keyframe selector g, i.e., $\boldsymbol{y} = [y_i] \in [0,1]^I$, we obtain the score of each shot by averaging the scores of the frames within the shot:

$$\bar{y}_m = \frac{1}{|\mathcal{I}_m|} \sum\nolimits_{i \in \mathcal{I}_m} y_i, \ \forall\, m = 1, ..., M. \tag{8}$$

where $\mathcal{I}_m$ is the set of frames within the m-th shot, and $|\mathcal{I}_m|$ indicates the number of frames within the m-th shot.

Given the shot-level scores $\bar{\boldsymbol{y}} = [\bar{y}_m] \in [0,1]^M$, we can utilize the 0/1 Knapsack algorithm [47] to select key shots for the video summary, which corresponds to solve the following optimization problem:

$$\boldsymbol{u}^* = \arg\max\nolimits_{\boldsymbol{u} \in \{0,1\}^M} \langle \boldsymbol{u}, \bar{\boldsymbol{y}} \rangle, \quad \text{s.t.} \langle \boldsymbol{u}, \boldsymbol{l} \rangle \leq L, \tag{9}$$

where $\boldsymbol{u}^* = [u_m]$ is a binary vector indicating whether a shot is selected or not. $\boldsymbol{l} = [|\mathcal{I}_m|] \in \mathbb{R}^M$ is a vector storing the length of the M shots. L is the pre-defined summary length. The problem maximizes the overall scores of the selected shots and ensures that the total number of their frames meets the constraint.

A.3 Implementation of the baselines

IV-Sum [35]. When implementing IV-Sum, we only use the cross-modal saliency scores since task annotations are unavailable for the WikiHow dataset, making IV-Sum an unsupervised method in this case. Besides, for a fair comparison, we use the shot-level textual descriptions generated by our method, rather than ASR texts and the corresponding descriptions provided in the WikiHow dataset.

CLIP-It [36]. When implementing CLIP-It, we follow the text generation approach described in the original paper [36]. Specifically, we generate dense paired textual descriptions for the *whole video* by applying the Bi-Modal Transformer (BMT) model [21].

B ADDITIONAL RESULTS

B.1 Robustness to Hyperparameters

We assess the robustness of hyperparameters by investigating the influence of the threshold K for alignment scores and the number of considered neighbors K' for representation scores. Both two parameters are determined by multiplying a factor by the total number of frames I within the input video. In our implementation, the K and K' are heuristically set to $0.8 \times I$ and $0.5 \times I$, respectively.

Figure 5 illustrates how the performance of our method changes with variations in the two parameters. Firstly, The K serves as the upper limit of considered frames for each text when calculating alignment scores. Setting a large value for K ensures the inclusion of all relevant frames when given a video with a low frequency of scene changes, e.g. the videos in the SumMe dataset. Secondly, the K' indicates the proportion of coherent scenes within the entire video by considering the reconstruction power of each frame with its neighboring K' frames. Increasing K' leads to an enhancement of model performance, indicating that the prevailing video distribution contains a greater proportion of contiguous scenes than the current K', and vice versa. As video distribution can vary for different datasets or data sources, these parameters need to be tailored accordingly to achieve optimal performance.

B.2 More Quantitative Results

In addition to the baseline methods listed in Table 1, we conduct further quantitative comparisons on the SumMe dataset, incorporating three more methods: unsupervised methods DSR-RL[43], AC-SUM-GAN [2], and supervised method MSVA [18]. For each of these methods, we employ the officially published implementation codes and keep the experimental parameters at their default settings, as provided in the codes. The comparative results are summarized in Table 8, where our method demonstrates superiority over these three baselines.

B.3 Metrics Comparison

Generic video summarization datasets accompany videos with multiple user annotations, but the quality of these annotations may vary. Even for the same video, different users may provide significantly different summary annotations. On the TvSum dataset, previous approaches usually compare the generated summaries

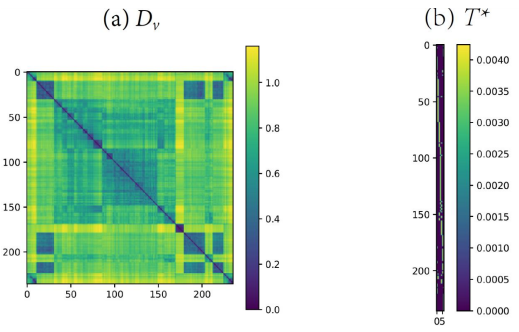

Figure 6: Visualization on distance matrix D_v and optimal transport matrix T^* between frames and textual descriptions.

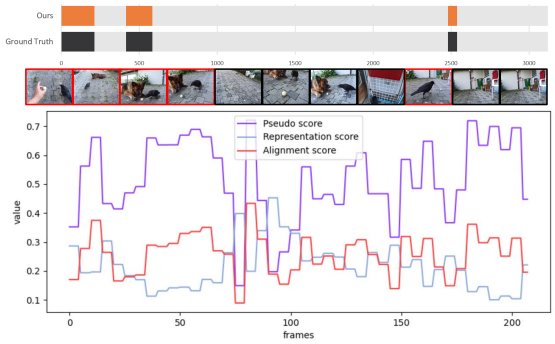

Figure 7: Comparison between ground-truth and machine-generated summary based on "video_25" in SumMe [19].

Figure 8: Comparison between ground-truth and machine-generated summary based on "Make-Coleslaw" in WikiHow [35].

Method	F1-Score	Precision	Recall
SUM-GAN [34]	77.5	77.4	77.5
DR-DSN [76]	78.5	78.6	78.5
CLIP-It [36]	81.5	81.3	81.7
Ours	85.7	85.6	85.8

Table 9: Ablation study of "MAX" metrics.

Method	Train	Test	Pseudo score	SUM
DR-DSN [76]	90	10	-	100
SUM-GAN [34]	4866	62	-	4928
CLIP-It [36]	28033	14	-	28047
Ours	299	49	123	471

Table 10: Comparisons on runtime (seconds) for various methods on SumMe dataset.

with each user summary and then average the multiple outputs to obtain the final results. Since the criteria for user summaries are not uniform, simply averaging metrics may introduce bias in the model's keyframe selection process by emphasizing obvious shots.

Therefore, we propose replacing the "AVG" metric with the "MAX" metric, which only requires that the generated summary sufficiently resembles a particular user summary rather than all user summaries. We compare our method with some state-of-the-art video summarization methods by using the 'MAX' metric on TVSum dataset, as depicted in Table 9. Our method surpasses all baselines in all three metrics, which suggests that the summary produced by our approach more closely aligns with a particular user summary and thus is of higher quality.

B.4 Runtime Comparison

When testing different methods, we observe a significant difference in their efficiency. We record the runtime of our approach alongside three baseline methods, each trained and tested 100 epochs on SumMe dataset. When recording the runtime of our method, we count the time spent on training the text projection module and the time spent on generating pseudo scores both in the "Pseudo score" column. According to Table 10, our approach is notably faster than SUM-GAN and CLIP-It. Specifically, our method is nearly 60 times more efficient in comparison to the CLIP-It approach. These results demonstrate that our proposed video summarization framework considerably reduces the complexity and improves the training efficiency by generating pseudo scores offline.

B.5 Visualization

Figure 6 displays distance matrix $\boldsymbol{D}_v = [d(\boldsymbol{v}_i, \boldsymbol{v}_j)] \in \mathbb{R}^{I\times I}$ and optimal transport matrix $\boldsymbol{T}^* \in \mathbb{R}^{I\times J}$ based on the "video_13" in TVSum. Through $\boldsymbol{D}_v$, we can learn about the similarity between frames in terms of content. Inspired by $\boldsymbol{D}_v$, we propose representation scores to pick out frames that carry more information. For each text, we can pick a small set of video frames corresponding to the text by the alignment scores generated according to OT matrix $\boldsymbol{T}^*$. Further, we can utilize the representation scores to distinguish and filter the video frames.

Figure 7 and Figure 8 provide qualitative comparisons between the ground-truth summary and the machine-generated summary for generic video summarization and instructional video summarization, respectively. We also plot the pseudo-importance scores, alignment scores, and representation scores corresponding to the example videos. The scores are downsampled for ease of comprehension and trend analysis. We can find that our alignment score accurately makes judgments about the importance of frames, with highlights being assigned high scores and some uneventful scenes being assigned low scores, which demonstrates the validity of our pseudo-supervision.