OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Exploring Contrastive Learning in Foundation Model Pre-training and Applications
Author(s)	Pang, Zongshang
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/103165
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Exploring Contrastive Learning in Foundation Model Pre-training and Applications

Submitted to

Graduate School of Information Science and Technology

The Univesity of Osaka

July, 2025

Zongshang PANG

Abstract

Contrastive learning, when scaled and adapted judiciously, offers a unifying objective for training *foundation models* that generalize from visual signals to multi-modal domains. This thesis shows how three complementary projects push that premise across granularity, supervision, and modality. In the first project, we propose a self-supervised pixel-level contrastive learning framework, *PixCon*. It investigates the effective components of current dense contrastive learning frameworks and systematically integrates them with a novel semantic reweighting mechanism, which enables simple pixel-level learning to outperform complex region-level approaches on dense visual prediction tasks such as detection and segmentation. In the second project, we propose a *training-free zero-shot video summarizer* by reformulating the classic diversity and representativeness video summarization heuristics as quantifiable scores based on contrastive losses, entirely eliminating task-specific fine-tuning while outperforming supervised baselines on TVSum and SumMe. In the third project, we investigate the potential of discriminative contrastive learning on generative models such as large language models on multi-modal video applications. Concretely, we propose the *S2L* framework that couples a video large language model that can perform a query-focused summarization task with a contrastive grounding module, transforming textual form summaries into precise timestamps and achieving new best results on ETBench localisation tasks. Collectively, these studies demonstrate that carefully engineered contrastive objectives can endow a wide spectrum of benefits on the pre-training and the applications of foundation vision and vision-language models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contrastive learning has served as the dominant self-supervised learning paradigm, yielding foundation models that achieve excellent performance across a broad spectrum of tasks and modalities [6–10]. Broadly speaking, the development and application of contrastive learning fall into three categories: (1) self-supervised representation learning for specific downstream tasks, (2) training-free zero-shot transfer to novel downstream tasks, and (3) auxiliary contrastive learning for supervised tasks.

The advent of the InfoNCE loss [11] has made contrastive learning the most effective self-supervised image representation learning approach. The primary contrastive learning treats images at the instance level: each image is mapped to a single feature vector; augmented views of the same image are pulled together, and features from different images are pushed apart. Although representations learned in this way have delivered excellent transfer performance on image classification benchmarks, many practical vision problems (*e.g.*, detection, segmentation) require richer, spatially aware features. To close that gap, several works have extended instance-level contrastive learning to dense prediction tasks. Pixel- or region-level contrastive frameworks [1, 12–15] adapt contrastive objectives so that spatially localized features (pixels or regions) are matched across views. In the vision-language domain, contrastive image-text learning methods such as CLIP [10] and SigLIP [16] have shown that pulling corresponding image and text pairs closer in a joint embedding space yields powerful, transferable features.

Region-level extensions of CLIP [17, 18] further adapt those representations to dense, multi-modal tasks. Contrastive image-representation learning has also been applied beyond the vision domain, for example, to audio classification [19] and medical images [20].

Contrastively learned representations often exhibit surprising emergent properties [9,21–23] that make training-free, zero-shot applications possible. For instance, DINO [9] showed that self-attention maps of a vision transformer trained with a contrastive-style objective reveal semantic segmentation patterns, even without any segmentation labels. Later work found that convolutional backbones trained with contrastive objectives exhibit similar localization cues [23]. Building on these observations, several methods propose training-free, zero-shot frameworks for semantic segmentation and object detection [24–27]. In the vision-language setting, CLIP's contrastive embeddings have been used to derive zero-shot text-conditioned segmentation and detection pipelines [18, 28, 29], and similar ideas have been extended to video applications [30–32].

Although contrastive losses were originally devised for self-supervised learning, they have also proven beneficial in (weakly) supervised learning contexts. Supervised contrastive learning [33] uses InfoNCE with positives defined by ground-truth labels. Beyond classification, InfoNCE has been incorporated as an auxiliary loss for semantic segmentation [34–37] and object detection [38–40]. In the video domain, contrastive losses often serve as auxiliary objectives to bolster retrieval or localization tasks [41–45]. Even in the era of large language models (LLMs), contrastive learning remains relevant: BLIP2 [46] uses contrastive vision-language pretraining to initialize a "Q-Former" that bridges a visual encoder to an LLM, and LLM2Vec [47] employs contrastive learning to convert generative LLMs into discriminative text encoders.

Therefore, contrastive learning's versatility, across self-supervised representation learning, training-free zero-shot transfer, and auxiliary supervised objectives, has spurred its application in tasks spanning multiple modalities, domains, and settings. In this thesis, we present three projects that respectively address each of the above categories of contrastive learning.

In Chapter 2, we revisit contrastive learning adapted for pixel-level pre-training and introduce PixCon, a framework that strengthens existing pixel-level baselines and rivals, or out-

performs, state-of-the-art region-level methods on dense prediction benchmarks. PixCon enhances "dense" InfoNCE objectives by aligning its training pipeline with recent advances in momentum-based contrastive frameworks (*e.g.*, MoCo-v2+ or BYOL), and by carefully incorporating both semantic-similarity and coordinate-based matching. Through extensive experiments on COCO and Pascal VOC, we show that PixCon's pixel-level features transfer strongly to object detection, instance segmentation, and semantic segmentation. By focusing on self-supervised pretraining tailored to dense vision tasks, PixCon exemplifies how contrastive objectives can be engineered to produce spatially discriminative features that excel when fine-tuned on downstream tasks with limited or no labels.

Chapter 3 explores the training-free zero-shot applications of contrastive image features in the context of video summarization. Without relying on any video-specific annotations, we design a framework that leverages pretrained, contrastively learned features to perform zero-shot video summarization, *i.e.*, we generate concise summaries of uncurated videos without any additional training on annotated summarization data. Our method formulates three complementary metrics (local dissimilarity, global consistency, and feature uniqueness) in the contrastive embedding space to rank and select representative frames. By clustering frames via these contrastive signals, we identify key moments that capture both per-sample distinctiveness and overall narrative coherence. Experiments on standard benchmarks (*e.g.*, SumMe, TVSum) demonstrate that our zero-shot summarizer matches, or sometimes surpasses, fully supervised methods, highlighting how pretrained contrastive embeddings can be harnessed directly for novel, downstream tasks.

In Chapter 4, we focus on leveraging contrasitive learning to facilitate LLM-based video temporal localization models, for which matching free-form text queries to specific video segments is essential. We introduce S2L, a framework that uses a contrastive context-matching module as an auxiliary objective to sharpen video-text alignment. Concretely, given a user query and a long video, a video LLM first generates a query-focused textual summary of the video. To localize the relevant segment, we train a contrastive matcher that aligns the summary's embedding to frame-level video features, effectively "pulling" the correct segment close

to the query-focused summary while "pushing" away irrelevant segments. By incorporating this contrastive loss alongside the conventional generative objective, S2L consistently improves localization accuracy on standard benchmarks. This work exemplifies how contrastive learning can serve as an auxiliary signal, complementary to the main generative modeling paradigm, to refine multimodal grounding with video LLMs.

Taken together, these three projects illustrate the breadth of contrastive learning's impact: from devising new self-supervised pretraining recipes for dense vision tasks (Chapter 2), to enabling zero-shot video analytics without any additional labels (Chapter 3), to acting as a complementary alignment objective in supervised multimodal systems (Chapter 4).

Chapter 2

PixCon: Pixel-Level Contrastive Learning Revisited

2.1 Overview

Contrastive image representation learning [6–9, 21, 48–50], which pulls closer the features of positive pairs produced by applying data augmentation to the same image while maximizing the distance between the features of negative samples, greatly advances the transfer learning performance of vision foundation models. Instance discrimination [48] methods work with global average-pooled image feature vectors and are thus referred to as *image-level learning* methods [1, 14, 51, 52]. Such methods are highly effective in improving models' image classification performance but often struggle to improve their performance on dense prediction tasks such as object detection [53] and semantic segmentation [54]. Various researchers propose to generalize image-level contrastive learning to work with dense spatial image features to facilitate transfer learning to dense prediction tasks [1, 13–15, 51, 52, 55–57]. Therefore, such methods are usually referred to as *dense learning* methods due to their focus on dense spatial features.

Though image-level learning methods are highly effective when applied on instance-centric images, *e.g.*, ImageNet [58], they are less promising in pre-training with scene-centric images

with multiple instances and complex structures [14, 15, 51, 59], such as MS COCO images [60]. To better utilize scene images during the contrastive pre-training of vision foundation models, *pixel-level* [1, 61] and *region-level* [13–15, 51, 52, 55–57] methods have been proposed. Pixel-level learning works with individual spatial feature vectors, whereas region-level learning works with selective aggregations of them. To construct positive pairs for pixel-level learning, the semantically closest spatial feature vectors [1] in the two respective views are used. Region-level methods consider this to be insufficient for exploiting complex scene structures and leverage various region-mining algorithms, such as unsupervised object detection [8, 50, 62, 63] or segmentation [9, 64], to obtain regions of interest for constructing region-level positive pairs. A conceptual illustration of their positive matching processes is provided in Figure 2.1.

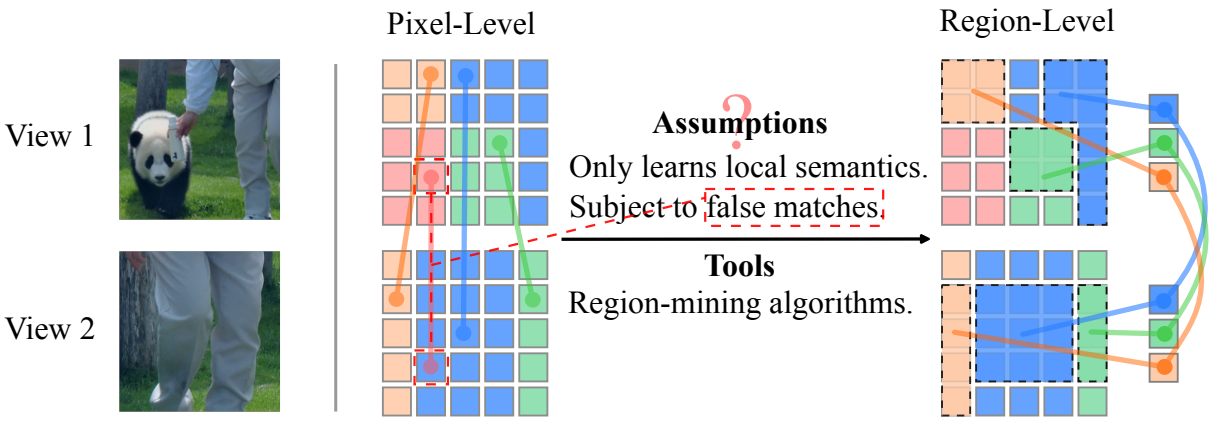

Figure 2.1: An illustration of the common assumptions regarding the differences in pixel and region-level learning methods. Girds' colors roughly indicate pixels' associated semantic classes based on the two input views for illustration purposes. The cross-view pixels connected by solid lines with round markers indicate positive matches. The matching process for pixel-level learning imitates the similarity-based matching from [1]. Region-level methods are motivated by the shown assumptions about pixel-level learning and rely on region-mining algorithms as tools to perform learning based on regional features. In this paper, we question these assumptions about pixel-level learning and revisit it to further exploit its potential. © [2024] IEEE.

Moreover, the random cropping step used to create positive pairs for performing contrastive learning risks creating semantically inconsistent views, which causes features with different semantic meanings, *e.g.*, different objects or objects and backgrounds, to be correlated. An example of such cases is provided in Figure 2.1, where the panda only appears in the first view but will be forced to correlate with the human's features by contrastive learning. With the help of region-mining algorithms, region-level methods are usually considered to be better at handling such cases, as they can rely on unsupervised region masks to evaluate the semantic consistency between the views.

In the conference version of this paper [65], we primarily revisited pixel-level learning and showed that (1) the potential of the pixel-level learning baseline, DenseCL [1], has not been fully exploited; (2) regional semantics can also emerge by applying pixel-level learning; and (3) pixel-level learning readily provides tools to successfully address the problem of semantically inconsistent scene crops. Specifically, this paper makes the following contributions:

- We propose *PixCon*, A stronger pixel-level contrastive learning framework, which augments DenseCL [1] by aligning its training pipeline with that of state-of-the-art (SOTA) region-level methods [14, 15, 51, 52, 57, 66]. We show that PixCon outperforms SOTA region-level methods in terms of transfer learning tasks.
- We thoroughly analyze pixel-level learning based on two positive matching schemes: semantic similarities [1] and spatial coordinates [14, 61]. We name the corresponding models *PixCon-Sim* and *PixCon-Coord*. We show that the similarity-based scheme intrinsically encourages the learning of regional semantics that region-level methods focus on.
- Finally, we propose *PixCon-SR* with a *semantic reweighting* strategy to deal with semantically inconsistent scene crops by jointly utilizing spatial and semantic information. PixCon-SR achieves better or competitive transfer performance compared with current SOTA methods on dense prediction tasks, including PASCAL VOC object detection [67], COCO object detection and instance segmentation [60], PASCAL VOC semantic seg-

mentation [67], and Cityscapes semantic segmentation [68].

In the journal version of this project [69], we provide further analyses of PixCon:

- We provide a detailed analysis of how each new component in PixCon's training pipeline contributes to improving DenseCL's performance to match that of region-level methods.
- As pixel-level learning frameworks rely on an additional image-level loss to work well, we add it to region-level methods for a fairer comparison. We show that the region-level methods cannot leverage the image-level loss.
- We show that there exist challenges to improving region-level methods with pixel-level matching strategies, which opens new opportunities for future research toward more robust, dense contrastive learning frameworks.

2.2 Related Work

Image-level Self-Supervised Learning. Pretext tasks such as predicting colors [70], relative positions [71], or the rotations of pixels [72] are essential to self-supervised image representation learning. Instance discrimination [48] based on contrastive learning has recently become the most effective pretext, where augmented views of the same image are drawn closer to one another and pushed farther from different images [6,7,50]. Though both the pulling and pushing forces are proven to be essential in contrastive learning [21], BYOL [8] came up with techniques to only optimize the pulling part of contrastive loss.

As the aforementioned methods invariantly treat each image as a single feature vector, they are referred to as *image-level learning* methods. Though the resulting models excel at image classification, they perform less impressively in transferring to dense prediction tasks, which rely on sufficiently discriminative spatial features, which image-level methods do not explicitly optimize.

Dense Self-Supervised Learning. By directly optimizing spatial image features, dense learning methods yield better transfer performance in dense prediction tasks. Among them,

pixel-level methods rely on crafting cross-view pixel-level positive matches utilizing either spatial coordinates [61] or bootstrapping semantic similarities [1]. As such pixel-level methods are considered insufficient for leveraging the rich semantics in complex scene images, *region-level* methods rely on region-mining algorithms, such as unsupervised object region proposal methods [8, 50, 62, 63], used by [15, 51, 55], or unsupervised segmentation algorithms [64, 73], used by [57, 66], to find semantically meaningful regions, which are then used to aggregate spatial features for contrastive [50] or self-distillation learning [8]. Additionally, [14] and [52] utilize learnable prototypes to perform unsupervised segmentation, while PixPro [13] relies on spatial distances to select semantically related features. However, we will show that region-mining algorithms are not as crucial to mining regional semantics as claimed for current region-level methods, as pixel-level learning methods can also be exploited to promote region-level learning.

Learning with Scene-Centric Images. The complex structures of scene-centric images, such as those from MS COCO [60], often cause challenges to the fundamental positive pair creation strategy, *i.e.* siamese learning with two augmented image views. Specifically, random crops of multi-object scene images may include totally different objects, and pulling their features closer does not contribute to learning semantically meaningful features. Region-level methods that rely on object proposals or segmentation masks can roughly evaluate the semantic consistency of the positive pairs and thus largely avoid such a problem, though at the cost of complicated pre-processing [15, 59, 66, 74], nontrivial computational burden during training [57], or less transferable features [14, 52] compared with pixel-level methods. However, we will show that tools to alleviate the negative influence of semantically inconsistent videos can be crafted with pixel-level learning alone.

2.3 Preliminaries

This section reviews two popular image-level learning pipelines, MoCo-v2 [7] and BYOL [8], where the latter is the default pipeline of most region-level methods. We also introduce a variant of MoCo-v2 with a similar architecture to that of BYOL, coined MoCo-v2+ by [75].

Common to MoCo-v2 and BYOL, each input image is augmented into two different views, $\mathbf{x}_1 \sim \mathcal{T}_1(\mathbf{x})$ and $\mathbf{x}_2 \sim \mathcal{T}_2(\mathbf{x})$, which are then fed into the *online* encoder f_θ and the *target* encoder f_ξ, where θ represents the learnable parameters and ξ is the exponential moving average of θ. The encoders are backbone networks, *e.g.*, ResNet [76], appended with two-layer multilayer perceptions (MLPs). The MLPs are usually called *projection heads*. The f_θ in BYOL has an additional two-layer MLP called the *predictor*, resulting in an asymmetric structure between the two encoders. Moreover, MoCo-v2 feeds each view into either the online or the target encoder to compute a loss $\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2)$, while BYOL sends each view to both encoders and symmetrizes the loss computation with respect to the two views, *i.e.* $\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}_{\text{img}}(\mathbf{x}_2, \mathbf{x}_1)$. Huang et al. [75] added, to MoCo-v2, the asymmetric encoder structure, where the online encoder contains a predictor, and the symmetrized loss, with

$$\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) = -\log \frac{\exp(\mathbf{q}\cdot\mathbf{k}^+/\tau)}{\sum_{\mathbf{k}\in\{\mathbf{k}^+\}\cup\mathcal{K}} \exp(\mathbf{q}\cdot\mathbf{k}/\tau)}, \tag{2.1}$$

where $\mathbf{q} = f_\theta(\mathbf{x}_1)/\|f_\theta(\mathbf{x}_1)\|_2$ is the query feature and $\mathbf{k}^+ = f_\xi(\mathbf{x}_2)/\|f_\xi(\mathbf{x}_2)\|_2$ is the positive key feature. $\mathcal{K}$ is the set of f_ξ outputs from other images which are $\mathbf{q}$'s negative key features stored in a fixed-length queue [7], and τ is the temperature coefficient. $\mathcal{L}_{\text{img}}(\mathbf{x}_2, \mathbf{x}_1)$ is computed by obtaining the query from $\mathbf{x}_2$ and the positive key from $\mathbf{x}_1$. The loss in Equation (2.1) is usually referred to as the InfoNCE loss [11]. In contrast, BYOL only aligns the positive features by maximizing their cosine similarities [8].

Additionally, BYOL also applies a momentum ascending strategy for updating ξ and synchronized batch normalization [77] as opposed to shuffling batch normalization [7] in MoCo-v2. When MoCo-v2 is equipped with these BYOL-style designs, it is called MoCo-v2+ in [75], demonstrating similar linear probing and transfer learning performance to those of BYOL but better than those of MoCO-v2. Moreover, SimSiamese [78] is a simplified version of BYOL, achieving better performance under similar training settings. For simplicity, we refer to BYOL, MoCo-v2+, and SimSiamese as BYOL pipelines if not stated otherwise.

2.4 Proposed Method

Based on MoCo-v2+, we add another asymmetric prediction structure to the backbone that outputs dense spatial feature maps, or *pixel-level* (*pixels*, in this context, refers to spatial components of dense feature maps as opposed to those of the input RGB images) features [14,51,52]. The online encoder f_θ now gives two sets of feature vectors, $\mathbf{q} \in \mathbb{R}^C$ and $\mathbf{U} \in \mathbb{R}^{S^2 \times C}$ (after flattening the first two dimensions), where C is the feature dimensionality and S denotes the length and width of the dense feature maps, which are set as equal for simplicity. Similarly, the target encoder f_ξ gives $\mathbf{k} \in \mathbb{R}^C$ and $\mathbf{V} \in \mathbb{R}^{S^2 \times C}$. Figure 2.2 provides a schematic illustration of the forward process. Based on this forward pipeline, we propose different variants of a pixel-level contrastive learning framework, namely, *PixCon*, with the loss function being

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = \mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2), \tag{2.2}$$

where $\mathcal{L}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ is the pixel-level contrastive loss to be defined. The final loss is symmetrized with respect to the two views, *i.e.* $\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}(\mathbf{x}_2, \mathbf{x}_1)$.

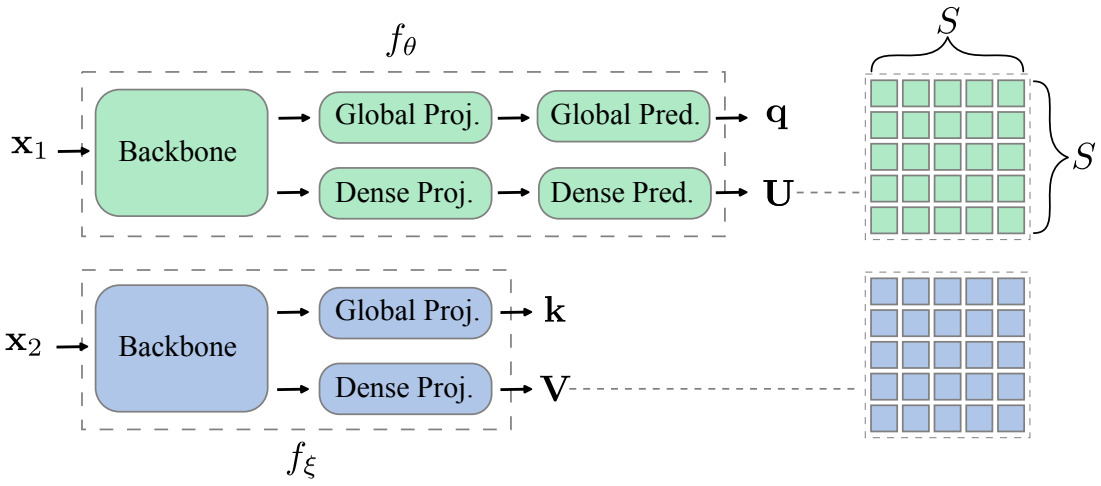

Figure 2.2: Both the online and the target encoders output two sets of outputs: global image-level outputs $(\mathbf{q}, \mathbf{k})$ and dense outputs $(\mathbf{U}, \mathbf{V})$. The dense outputs are of size $S \times S \times C$ before flattening the spatial dimensions. We leave out the visualization of global features and dense features' last dimension (C). © [2024] IEEE.

2.4.1 PixCon-Sim

Let the backbone networks' outputs be $\mathbf{F} \in \mathbb{R}^{S^2 \times C}$ and $\mathbf{F}' \in \mathbb{R}^{S^2 \times C}$ for the query and the key views, respectively; the spatial positions of the features in $\mathbf{F}$ are matched to those in $\mathbf{F}'$ by

$$l(i) = \arg\max_j sim(\mathbf{F}(i), \mathbf{F}'(j)), \tag{2.3}$$

where $i, j \in [0, S^2 - 1]$ and $sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b} / \|\mathbf{a}\| \|\mathbf{b}\|$. The similarity-based matching scheme aims to bootstrap feature similarities, *i.e.* features with better semantic correlation give more semantically meaningful matches, which are in turn used to strengthen the correlation of such features. Similar bootstrapping strategies are also applied in region-level methods [13, 14, 52, 57].

With similarity-based matching, the pixel-level contrastive loss is then computed as follows:

$$\mathcal{L}^l_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2) = -\frac{1}{S^2} \sum_i \log \frac{\exp(\mathbf{u}_i \cdot \mathbf{v}^+_{l(i)} / \tau)}{\sum_{\mathbf{v} \in \{\mathbf{v}^+_{l(i)}\} \cup \mathcal{V}} \exp(\mathbf{u}_i \cdot \mathbf{v} / \tau)}, \tag{2.4}$$

where $\mathbf{u}_i = \mathbf{U}[i] \in \mathbb{R}^C$, $\mathbf{v}^+_{l(i)} = \mathbf{V}[l(i)] \in \mathbb{R}^C$, and $\mathcal{V}$ contains image-level negative key features from other images, in accordance with [1], for computational efficiency. The negative keys are stored in a fixed-length queue.

However, the matching function in Equation (2.3) hardly makes sense at the beginning of training. As demonstrated in DenseCL [1], jointly conducting image-level and pixel-level learning can help mitigate the problem, as image-level learning also encourages the emergence of semantic relations among spatial features [23, 51]. Additionally, image-level learning is also commonly conducted along with dense learning [13, 15, 51] and brings benefits. Therefore, by using $\mathcal{L}^l_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Equation (2.2) and symmetrizing the resulting loss with respect to the two views, we obtain the final loss for *PixCon-Sim*, *i.e.* pixel-level contrastive learning with similarity-based matches. When using the MoCo-v2 pipeline instead of MoCo-v2+ and not using the symmetrized loss, PixCon-Sim becomes DenseCL [1].

2.4.2 PixCon-Coord

Though similarity-based matching gives increasingly better matches as the training proceeds [1], it still retrieves semantically inconsistent matches, especially at the beginning of training. To further investigate its pros and cons, we compare it with the coordinate-based matching scheme [13, 14, 61], which matches two cross-view spatial features only if they have (approximately) the same coordinates when mapped back to the input image space, thus guaranteeing semantic consistency among the positive matches.

Therefore, we propose another variant of PixCon using *coordinate-based* matching based on inverse augmentation [14], which involves RoIAlign [2] and horizontal flipping if the input image has been flipped. The schematic illustrations of both similarity-based matching and coordinate-based matching are provided in Figure 2.3.

By slightly overloading the notations $\mathbf{U}$ and $\mathbf{V}$ as the pixel-level outputs of inverse augmentation, we have the corresponding pixel-level loss $\mathcal{L}^c_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$, which replaces the matching function l in Equation (2.4) with c, which is defined as

$$c(i) = i, \tag{2.5}$$

connecting the same positions in the two views' feature maps aligned by inverse augmentation. By using $\mathcal{L}^c_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Equation (2.2) and symmetrizing the resulting loss with respect to the two views, we obtain the final loss for *PixCon-Coord*, *i.e.* pixel-level contrastive learning with coordinate-based matches.

2.4.3 PixCon-SR

As shown in Figure 2.3, the two augmented views of the input multi-object image are semantically inconsistent, *i.e.* the panda only appears in the first view. Thus, similarity-based matches for such view-specific objects' pixels will have different semantic classes. While coordinate-based matching helps mitigate such false matches, it only matches cross-view pixel-level features at (approximately) the same spatial location in the input image. As a result, it fails to relate

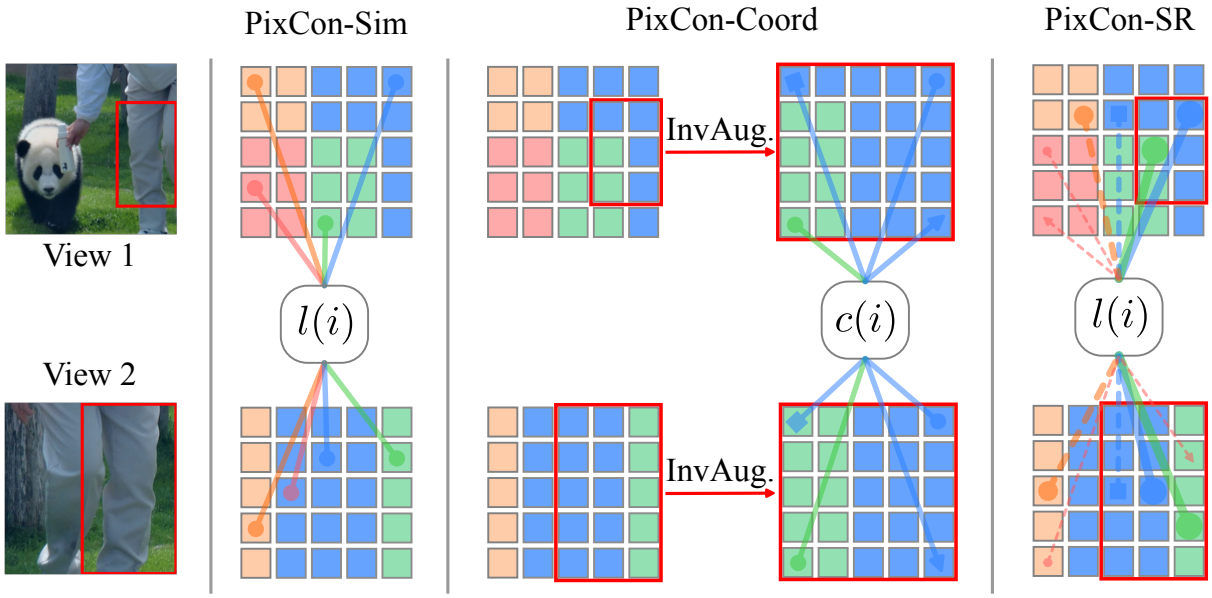

Figure 2.3: An illustration of different PixCon variants' matching schemes. The red bounding boxes indicate the intersected area of the two views. Girds' colors roughly indicate pixels' associated semantic classes for illustration purposes. We treat `view 1` as the `query` view and `view 2` as the `key` view. PixCon-Sim's matching scheme is the similarity-based matching in Equation (2.3). PixCon-Coord uses the matching function in Equation (2.5), and the involved inverse augmentation includes RoIAlign [2] and optional horizontal flipping depending on whether the input is flipped. PixCon-SR uses similarity-based matching but applies the semantic reweighting in Equation (2.7). For the illustration of PixCon-SR, solid lines indicate matches with `query` pixels in the red bounding box, dashed lines represent the rest of the matches, and different line widths indicate the magnitudes of semantic weights. The matches are drawn for illustration purposes, and not all are drawn for clarity. © [2024] IEEE.

semantically related but spatially distant features, whereas pulling such features closer is crucial to learning regional semantics for better transfer performance [14, 15, 51, 66]. Therefore, it is natural to ask the following question: how do we leverage the benefits of both similarity- and coordinate-based matching schemes?

We start to craft a matching scheme that leverages both spatial and semantic information by further noting the hidden problems of similarity-based matching. Firstly, some matches with low similarities can actually be highly semantically close, constituting hard positive pairs that are important to leverage for better feature quality [79]. Moreover, although similarity-based positive matches share maximal similarities among cross-view samples, the similarities can still be low, indicating that they belong to different semantic classes. To exploit hard positive samples, we choose to fully trust positive matches whose query pixels lie in the intersection of two views regardless of the query-key similarity. We call such queries the "in-box" queries, as the intersection area is always a box. The matched key for an in-box query is highly likely to be meaningful, as the query is guaranteed to have semantic correspondences in the key view, *e.g.*, the same pixel itself in the key view in the worst case. To address the negative influence of positive matches with low matching similarities, we propose to reweight such matches with "out-of-box" queries by their query-key similarities. We illustrate such a reweighting process in Figure 2.3.

We term the consequent reweighting strategy *semantic reweighting*, with which the pixel-level loss becomes

$$\mathcal{L}_{\text{pix}}^{l,w}(\mathbf{x}_1, \mathbf{x}_2) = -\sum_i \frac{w(i)}{A} \log \frac{\exp(\mathbf{u}_i \cdot \mathbf{v}_{l(i)}^+ / \tau)}{\sum_{\mathbf{v} \in \{\mathbf{v}_{l(i)}^+\} \cup \mathcal{V}} \exp(\mathbf{u}_i \cdot \mathbf{v} / \tau)}, \tag{2.6}$$

where $A = \sum_i w(i)$ is the normalization factor. Let $\mathcal{Y}$ be the set of indices of the in-box query features, which can be easily obtained during data augmentation; we compute $w(i)$ as

$$w(i) = \begin{cases} 1, & \text{if } i \in \mathcal{Y}. \\ norm(\max_j sim(\mathbf{F}(i), \mathbf{F}'(j)))^\alpha, & \text{otherwise.} \end{cases} \tag{2.7}$$

where $norm(x) = (x - \min_{j \notin \mathcal{Y}} w(j))/(\max_{j \notin \mathcal{Y}} w(j) - \min_{j \notin \mathcal{Y}} w(j))$ guarantees the continuity of weights and enlarges their contrast and α is for further sharpening the contrast and is set to 2 by default. Note that the formulation of Equation (2.6) is not related to inverse augmentation, which is more computationally expensive, *i.e.* $\mathbf{U}$ and $\mathbf{V}$ are dense outputs from f_θ and f_ξ. By using $\mathcal{L}_{\text{pix}}^{l,w}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Equation (2.2) and symmetrizing the resulting loss with respect to the two views, we obtain the final loss for *PixCon-SR*, *i.e.* pixel-level contrastive learning with semantic reweighting.

PixPro [13] also simultaneously utilizes spatial information and feature similarities. However, they use spatial information to retrieve positive matches, whose quality highly depends on the pre-defined size of a spatial neighborhood. We impose no spatial constraint on the positive matches at all and only bootstrap feature similarities. Due to the use of spatially close positive matches, they need to use self-attention maps to relate spatially distant pixels, whereas we merely rely on pixel-level features together with default random cropping and the inherent uncertainty of similarity-based matching to achieve this purpose.

2.5 Experiments

2.5.1 Experimental Settings

Datasets. For pre-training, as we are mainly interested in pre-training on real-world scene images containing diverse and complex contents, we use the training set of MS COCO [60], which contains $\sim$118k images and is broadly used for scene-level pre-training. COCO is also widely used for benchmarking dense prediction tasks such as object detection, instance segmentation, and semantic segmentation. Moreover, a COCO image contains 7.3 objects on average, which is in stark contrast to the meticulously curated ImageNet [58] images, for which the number of objects per image is 1.1 [1].

Architecture. We base our architecture on that of MoCo-V2+ [75]. Following [1], we add dense learning branches to the global learning branches. Specifically, the online encoder

has a ResNet50 [76] backbone, which is appended with a global projection head and a dense projection head. The former has two fully connected layers, while the latter has two 1×1 convolutional layers. Both heads have batch normalization followed by ReLU in between the two layers. For both heads, the hidden dimensionality and the output dimensionality are 2048 and 128, respectively. The global and dense heads are appended with their respective predictors, which have the same architectures as the heads with an input dimensionality of 128. The target encoder has the same architecture as the online encoder except that it does not have predictors.

Data augmentation. Pre-training data augmentation is in accordance to [8], where each image is randomly cropped into two views, which are then resized to 224×224, followed by random horizontal flipping, color distortion, Gaussian blur, and solarization. Crops without overlapping are skipped.

Pre-training setup. Following [1], the negative-storing queues for both global learning and dense learning are of length 65,536. The momentum for updating the target encoder is initially set to 0.99 and increased to 1 at the end of training [8]. Synchronized batch normalization [77] is used for all batch normalization layers [8]. The temperature τ is set to 0.2. We use the SGD optimizer with an initial learning rate of 0.4 and a cosine learning rate decay schedule. We set the weight decay to 0.0001 and the momentum for the optimizer to 0.9. We train each model for 800 epochs on COCO with four GPUs and a total batch size of 512. Training is conducted under the MMSelfSup framework [80].

Evaluation settings. We follow previous work [1, 6, 7, 14, 15] to evaluate feature transferability by fine-tuning the pre-trained models on target downstream tasks. We then evaluate the resulting models by reporting the metrics used in the corresponding tasks, including VOC object detection [67], COCO object detection, COCO instance segmentation [60], VOC semantic segmentation [67], and Cityscapes semantic segmentation [68].

For VOC object detection, we fine-tune a Faster R-CNN with a C4-backbone. Training is performed on the VOC `trainval07+12` set for 24k iterations. The evaluation is performed on the VOC `test2007` set. Both training and evaluation use the Detectron2 [81] code base.

For COCO object detection and instance segmentation, we fine-tune a Mask R-CNN with

an FPN backbone on COCO's `train2017` split with the standard $1\times$ schedule and evaluate the fine-tuned model on COCO's `val2017` split. Following previous work, we synchronize all the batch normalization layers. Detectron2 is used to conduct the training and evaluation.

We strictly follow the settings in [14] for VOC and Cityscapes semantic segmentation. Specifically, an FPN is initialized with the pre-trained model, fine-tuned on the `train_aug2012` set for 30 k iterations, and evaluated on the `val2012` set. For Cityscapes, we conduct fine tuning on the `train_fine` set for 90 k iterations and evaluate the fine-tuned model on `val_fine`. The training and evaluation are conducted by using MMSegmentation [82].

The results, including ours and those of reproducible previous methods, are reported as the average of five, three, three, and five independent runs for VOC detection, COCO detection and instance segmentation, Cityscapes segmentation, and VOC segmentation, respectively.

2.5.2 Main Results

As discussed in Section 2.4.1, PixCon-Sim boils down to DenseCL [1] when not applying the BYOL pipeline; this is, however, invariantly used by the region-level methods. . As per Table 2.1, PixCon-Sim outperforms DenseCL across all the benchmarks. Additionally, with a simple pixel-level learning algorithm, PixCon-Sim is already competitive compared with region-level methods across all the benchmarks. PixCon-Coord, with a geometric matching scheme, is also competitive.

For all four tasks, PixCon-SR brings consistent performance boosts to its image-level baseline MoCo-v2+ and surpasses previous region-level methods, as well as the other two PixCon variants. Though PixCon-SR's performance on COCO detection and instance segmentation is similar to that of UniVIP [15] and SlotCon [14], it has better performance in terms of the other three tasks. It achieves this without relying on any region-mining algorithms, as shown in Table 2.2, most of which resort to complex pre-processing or computationally expensive multi-stage training. Specifically, for prototype-based methods, *i.e.* DenseSiamese [52] and SlotCon [14], their transfer performance in VOC detection is conspicuously lower than that of the other methods. This is likely caused by the fact that the dense features are trained to

Table 2.1: Main transfer results. All self-supervised models were pre-trained for 800 epochs on COCO, except that DetCon was trained for 1000 epochs. Among all the methods, MoCo-v2 and DenseCL are based on the MoCo-v2 pipeline, while the others are based on the BYOL pipeline. Refer to Section 2.4 for more details on the differences between the pipelines. We also categorize the methods into different types based on their training strategies, including image level, region level, and pixel level. Refer to Table 2.2 for more information about region- and pixel-level methods. On all the benchmarks, our method shows strong transfer performance. We use boldface to indicate single best results but underline multiple best results that have the same value (†: re-impl. w/official weights. ‡: full re-impl.). © [2024] IEEE.

Method	**Type**	**VOC Detection**			**COCO Detection**			**COCO Instance seg.**			**City. Seg.**	**VOC Seg.**
		AP	**AP$_{50}$**	**AP$_{75}$**	**AP^{bb}**	**AP$^{bb}_{50}$**	**AP$^{bb}_{75}$**	**AP^{mk}**	**AP$^{mk}_{50}$**	**AP$^{mk}_{75}$**	**mIoU**	**mIoU**
Random init. [1, 14]	-	32.8	59.0	31.6	32.8	50.9	35.3	29.9	47.9	32.0	65.3	39.5
MoCo-v2 [50]	Image	54.7	81.0	60.6	38.5	58.1	42.1	34.8	55.3	37.3	73.8	69.2
BYOL ‡ [8]		55.7	81.8	61.6	39.5	59.4	43.3	35.6	56.6	38.2	75.3	70.2
MoCo-v2+ ‡ [75]		54.6	81.4	60.5	39.8	59.7	43.6	35.9	57.0	38.5	75.6	71.1
ORL † [51]	Region	55.8	82.1	62.3	40.2	60.0	44.3	36.4	57.4	38.8	75.4	70.7
PixPro [13]		-	-	-	40.5	60.5	44.0	36.6	57.8	39.0	75.2	72.0
DetCon [66]		-	-	-	39.8	59.5	43.5	35.9	56.4	38.7	76.1	70.2
UniVIP [15]		56.5	82.3	62.6	<u>40.8</u>	-	-	<u>36.8</u>	-	-	-	-
Odin ‡ [57]		56.9	82.4	63.3	40.4	60.4	44.6	36.6	57.5	39.3	75.7	70.8
DenseSiam [52]		55.5	81.1	61.5	-	-	-	-	-	-	-	-
SlotCon † [14]		54.5	81.9	60.3	<u>40.8</u>	<u>61.0</u>	<u>44.8</u>	<u>36.8</u>	**58.0**	39.5	76.1	71.7
DenseCL [1]	Pixel	56.7	81.7	63.0	39.6	59.3	43.3	35.7	56.5	38.4	75.8	71.6
PixCon-Sim (ours)		57.3	82.4	63.9	40.5	60.5	44.2	36.6	57.5	39.2	76.1	72.6
PixCon-Coord (ours)		57.2	82.6	63.4	40.3	60.3	43.9	36.5	57.4	39.2	75.8	72.3
PixCon-SR (ours)		**57.6**	**82.8**	**64.0**	<u>40.8</u>	<u>61.0</u>	<u>44.8</u>	<u>36.8</u>	57.9	**39.6**	**76.6**	**73.0**

Table 2.2: Comparisons between region- and pixel-level methods. While most of the region-level methods require object priors, multi-stage training, or prototype learning, pixel-level methods need none of them. © [2024] IEEE.

Method	Scheme	Obj. Prior	Multi-Stage	Proto.
ORL [51]	Region level	✓	✓	×
PixPro [13]		×	×	×
DetCon [57]		✓	×	×
UniVIP [15]		✓	×	×
Odin [57]		×	✓	×
DenseSiam [52]		×	×	✓
SlotCon [14]		×	×	✓
DenseCL [1]	Pixel level	×	×	×
PixCon-∗		×	×	×

cluster around a fixed number of prototypes, which may cause the features to be overfitted to the prototypes and thus may hurt the transfer performance due to overly small intra-class variances [83]. The pre-training based on a specific number of prototypes also struggles to serve multiple downstream tasks equally well [14]. Overall, Table 2.1 sufficiently indicates the potential of pixel-level learning and the effectiveness of PixCon-SR.

2.5.3 Detailed Analysis

Similarity-based matching encourages learning regional semantics. Compared with the similarity-based matching used for PixCon-Sim, the coordinate-based matching of PixCon-Coord guarantees semantic consistency between the positive matches, as the matches represent the same patch in the image, which undergoes different augmentations. However, such strict geometric matching does not encourage relating spatially distant pixels associated with the same object and is thus limited in learning regional semantics.

Though similarity-based matches do not always enjoy such geometric proximity, their semantic consistency becomes increasingly better as training proceeds if the query feature has semantic correspondences in the key view [1]. For query pixels not lying in the intersection of the two views, *i.e.* out-of-box queries, their matches in the key view are guaranteed to be spatially apart from them. When such matches are semantically related, they could strengthen the correlation of spatially distant pixels belonging to the same semantic group. A qualitative investigation in the form of self-attention maps is provided in Figure 2.4, where semantically related but spatially distant pixel features are more holistically correlated for PixCon-Sim than for PixCon-Coord and MoCo-v2+. Moreover, Table 2.1 shows that PixCon-Sim delivers better transfer performance compared with PixCon-Coord, which may be attributed to the better regional semantics made possible by the similarity-based matching.

Semantic reweighting helps learn better regional semantics. The semantic reweighting strategy of PixCon-SR in Section 2.4.3 aims to discount the influence of inaccurate matches caused by semantically inconsistent views of scene images while utilizing as many semantically consistent matches as possible. Therefore, we expect the resulting features to be less correlated when they are associated with different semantic classes and have better intra-class coherence. Indeed, Figure 2.4 shows that PixCon-SR's self-attention maps allow for a better localization of semantic objects compared with PixCon-Sim (less attention on features of different semantic classes) while guaranteeing sufficient coverage of whole objects (better intra-class cohesion), even when compared with the region-level method SlotCon [14]. Moreover, as shown in Table 2.1, PixCon-SR achieves better transfer performance compared with PixCon-Sim and PixCon-Coord, as well as previous region-level methods, which further indicates the efficacy of the semantic reweighting strategy in helping learn decent regional semantics crucial to better transfer performance. Figure 2.5 provides visualizations of the semantic weights for the query features, where we can observe that the semantic contents not shared by the two views are given small weights and out-of-box query pixels with semantic correspondences in the key view are assigned nontrivial weights.

Designs of semantic reweighting. In Equation (2.7), spatial information is used to fully

utilize matches with better guarantees for their semantic consistency regardless of their feature similarities, as their queries, *i.e.* in-box queries, are present in the two views' intersected part and thus always have semantic correspondences in the key view. Additionally, feature similarities are used to reweight the matches with out-of-box queries to diminish the effect of semantically inconsistent ones while exploiting those that are still informative. Table 2.3 allows for an examination of the impact of these two tools based on ablation studies.

Interestingly, when using similarity-based matches with in-box queries alone, PixCon-SR (Spa.) achieves slightly better performance than PixCon-Coord, which also merely utilizes matches having in-box queries but with coordinated-based matching. This indicates that similarity-based matching provides matches with sufficient semantic consistency. While only using either spatial information or feature similarities does not give apparent performance gain, combining them, *i.e.* PixCon-SR (full), offers immediate improvements in the transfer performance, indicating the importance of sufficiently leveraging informative positives and mitigating the influence of false positives simultaneously.

Effect of the sharpening factor α. As shown in Table 2.3, the sharpening factor α does not cause drastic fluctuations in transfer performance, but a value of 2 helps strike a good balance between detection and semantic segmentation tasks, which is then applied as the default value.

A step-by-step investigation from DenseCL to PixCon-Sim. After applying the MoCo-v2+/BYOL training pipeline, MoCo-v2-based DenseCL becomes PixCon-Sim, which delivers consistently better transfer performance. It is thus interesting to investigate which newly introduced component in the new pipeline is contributing to better transfer performance.

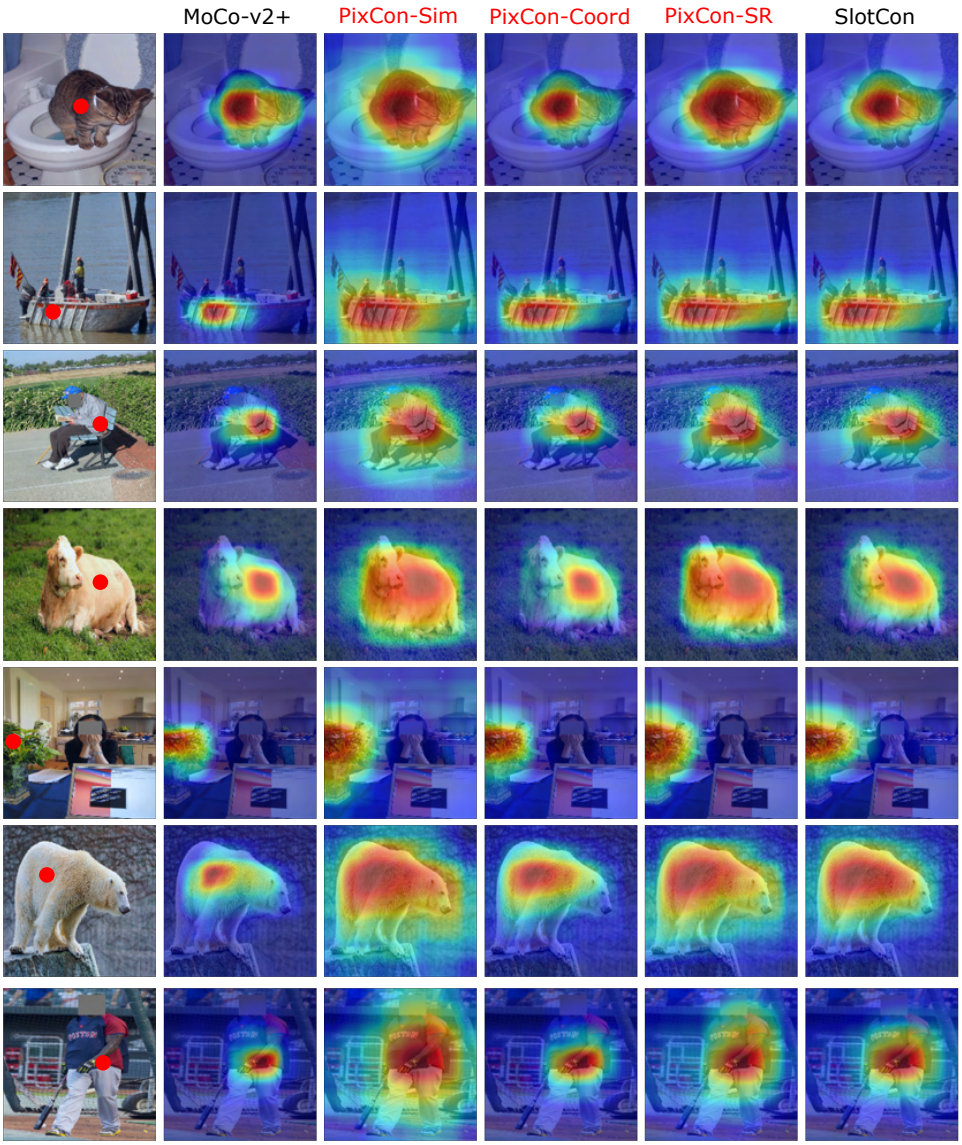

Figure 2.4: Visualizations of self-attention maps. For each row, the first image is the original image, with the red dot highlighting the pixel whose feature is used to calculate the cosine-similarity-based self-attention maps. The subsequent images are self-attention maps using different models' features. See the main texts for analyses. © [2024] IEEE.

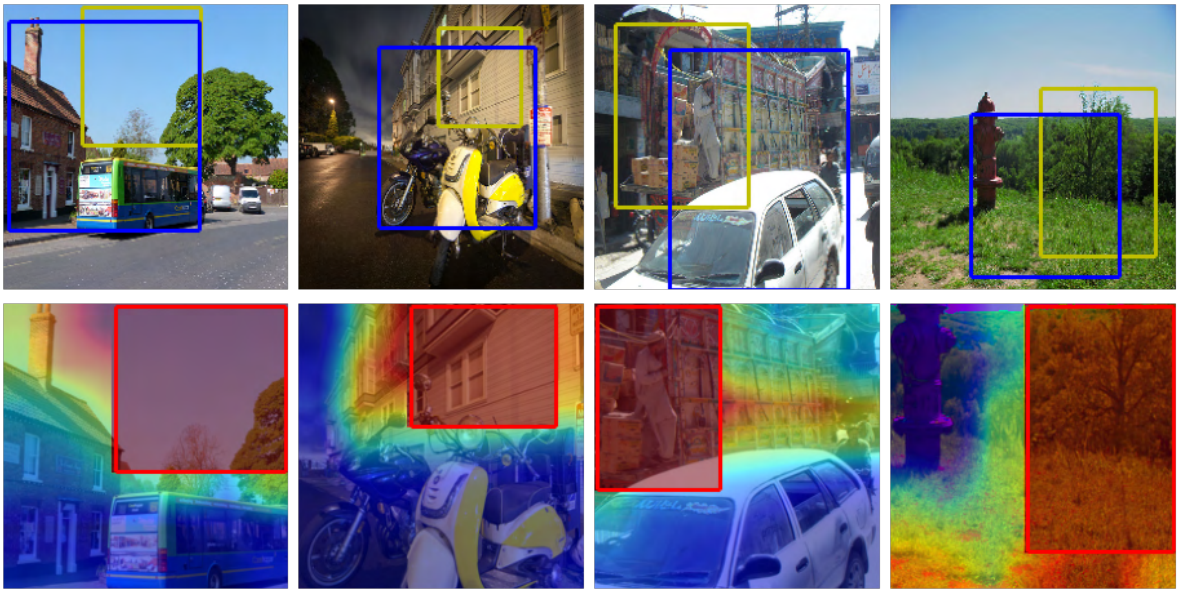

Figure 2.5: Visualizations of semantic weights. The first row shows the raw images with the blue bounding boxes indicating the query views and the yellow bounding boxes the key views. The second row shows the heatmap of semantic weights for the query pixels (in the blue bounding box), where the red bounding boxes indicate the intersection between query and key views. All images and heatmaps are resized to the same size for visualization purposes. © [2024] IEEE.

Table 2.3: We examine the influence of the tools used to formulate the semantic weights in Equation (2.7) based on ablation studies. *PixCon-SR (Spa.)* means that only matches whose query features lie in the two views' intersected parts are accepted and the other matches have weights 0. Here, only the spatial information is used for formulating the semantic weights. *PixCon-SR (Sim.)* means that only the similarities between the matched features are used as semantic weights, regardless of whether the query features exist in the two views' intersected area. *PixCon-SR (full)* utilizes both tools. The effect of the sharpening factor α in Equation (2.7) is also investigated here. © [2024] IEEE.

Method	α	**COCO**		**VOC Seg.**
		AP^{bb}	**AP^{mk}**	**mIoU**
PixCon-Sim	-	40.5	36.6	72.6
PixCon-Coord	-	40.3	36.5	72.3
PixCon-SR (Spa.)	2	40.5	36.5	72.5
PixCon-SR (Sim.)	2	40.3	36.4	72.3
PixCon-SR (Full)	2	40.8	36.8	73.0
PixCon-SR (Full)	1	40.5	36.5	73.2
PixCon-SR (Full)	4	40.5	36.6	73.0

Table 2.4: Investigating the effect of components in MoCo-v2+/BYOL on DenseCL's transfer performance. © [2024] IEEE.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
DenseCL	39.6	35.7	71.6
+ SyncBN	39.6	35.6	71.7
+ Asymmetric predictor	39.6	35.7	71.7
+ Momentum ascending	40.1	36.2	72.1
+ Symmetric loss	40.3	36.4	71.5
+ BYOL Aug. (**PixCon-Sim**)	40.5	36.6	72.6
− Symmetric loss	39.8	36.0	72.2

As shown in Table 2.4, SyncBN can be used to replace the ShuffleBN in MoCo-v2 without affecting transfer performance much. Asymmetric predictors do not have an apparent contribution. Momentum ascending, symmetric loss, and BYOL augmentation all contribute to better transfer performance, which is consistent with the observation made in the paper where MoCo-v2+ is introduced [75]. However, we found that symmetric loss and BYOL augmentation deliver a more consistent performance boost when applied together.

Though asymmetric predictors and SyncBN do not improve transfer performance, they have been shown, in [75], to contribute to linear probing accuracy on the pre-training dataset. If linear probing accuracy is not considered, it might be interesting to investigate the effect of removing these two techniques. However, to align with previous region-level methods, which invariantly incorporate all the BYOL components, we do so as well by default and leave the investigation for future work.

SlotCon and PixPro do not benefit from image-level loss. DenseCL [1] and the proposed PixCon framework both require image-level loss to work well. However, for the SOTA region-level methods, SlotCon [14] and PixPro [13], the former does not contain an image-level loss, while the latter does not use it by default. Therefore, we would like to investigate whether

Table 2.5: SlotCon and PixCon with image-level losses. © [2024] IEEE.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
SlotCon	40.8	36.8	71.7
SlotCon + image	40.5	36.6	70.2
PixPro	40.1	36.1	71.0
PixPro + image	40.5	36.6	69.8

an additional image-level loss will help these two methods. The experiments are based on the officially released codes of SlotCon and PixPro. As shown in Table 2.5, both SlotCon and PixPro fail to benefit from the additional image-level learning.

We can observe that all the reported methods have gained from leveraging more scene-centric images for pre-training. It is interesting to see that SlotCon has substantially better performance on VOC detection, COCO detection, instance segmentation, and VOC segmentation. UniVIP also witnessed an impressive performance boost on VOC detection after utilizing COCO+ for pre-training. PixCon-SR experienced consistent transfer performance improvements across the benchmarks and remains competitive compared with region-level methods. Interestingly, PixCon-SR falls behind SlotCon on ADE20k when pre-trained on COCO but catches up after COCO+ pre-training. SlotCon has a smaller relative improvement on ADE20k after pre-training on COCO+ compared with PixCon-SR.

Attempts to relax the use of prior knowledge in region-level learning. Among the region-level learning methods, there are two that also consider pixel-level features, *i.e.* PixPro and SlotCon. As opposed to pure pixel-level learning applied in DenseCL and the proposed PixCon, PixPro applies pixel-to-region matching based on self-attention to explicitly learn regional semantics. On the other hand, SlotCon enforces pixel-level features to be grouped under learnable prototypes, the number of which is tuned for them to capture region-level semantics. Additionally, SlotCon also applies an attention-based region-level loss. The common first

Table 2.6: Attempts to combine similarity-based matching with SlotCon. See text for analyses. © [2024] IEEE.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
SlotCon	40.8	36.8	71.7
SlotCon+Pix.	40.7	36.6	70.6
SlotCon-Coord.+Sim.	39.7	35.7	68.3
SlotCon-Coord.+Sim.+Img.	40.5	36.5	69.7
SlotCon+Sim.	40.5	36.6	69.5
SlotCon+Sim.+SR	40.7	36.7	70.5

step between pixel or pixel-to-region losses is to find pixel-level positive matches. DenseCL and PixCon find such matches mainly by bootstrapping feature similarities, while PixPro and SlotCon utilize a safer source of information based on prior knowledge, *i.e.* spatial coordinates.

As we have discussed in Section 2.5.3 in the main text, similarity-based matching encourages learning regional semantics more than coordinate-based matching. Thus, if we desire to learn regional semantics *without explicitly applying region-level learning*, similarity-based matching is the key. PixPro and SlotCon are equipped with coordinate-based matching, but they need to explicitly leverage region-level losses. One question that naturally comes to mind is the following: will similarity-based matching facilitate *explicit* region-level learning? In other words, we may want to know whether it helps to augment/replace the coordinate-based matching in PixPro or SlotCon with bootstrapping-driven similarity-based matching. We made several attempts in this direction but did not witness any improvements. The results are shown in Table 2.6. We provide our analyses of the results below.

SlotCon+Pix. means that we augment SlotCon with an additional pixel-level learning branch, for which we apply the PixCon pixel-level loss (without semantic reweighting). We can observe that simply augmenting SlotCon with similarity-based pixel-level learning does not help. *SlotCon-Coord.+Sim.* means that we replace coordinate-based matching with similarity-based

matching, and this scenario leads to a significant performance drop. This is expected, as similarity-based matching needs the image-level loss as a basis for semantically meaningful features, whereas SlotCon's region-level loss, similar to similarity-based matching, also relies on bootstrapping feature similarities. Therefore, the scenario *SlotCon-Coord.+Sim.+Img.*, where the image-level loss is added, shows more reasonable performance, which still does not match the original performance. Moreover, as shown in Table 2.5, SlotCon does not benefit from the image-level loss to begin with. When we tried to augment the original coordinate-based loss with the similarity-based loss on the same branch (*SlotCon+Sim.*), we observed a similar performance drop. Semantic reweighting (SR) helps regain part of the original performance. We observe similar trends for PixPro but only report SlotCon results here, as we have only managed to verify the reproducibility of SlotCon's code.

What could account for the failure? Compared with the straightforward pixel-level loss in PixCon, SlotCon, as well as PixPro, takes a step forward to further bootstrap feature similarities/attention for conducting region-level learning. Compared with similarity-based matching, which is already driven by bootstrapping, coordinate-based matching is apparently a safer tool for providing better semantically meaningful features, at least in the initial stage, to support such region-level bootstrapping. Semantic reweighting helps avoid part of the negative effect of bootstrapping by incorporating spatial information, but it still relies on similarity-based matching.

Similar to PixPro and SlotCon, the proposed PixCon framework is another step towards making dense representation learning less restricted by human prior knowledge via relying more on bootstrapping. Attempting to combine PixCon and region-level bootstrapping is yet another effort in the same direction but remains challenging for now and interesting for future work.

COCO+ results. To investigate whether PixCon-SR can further benefit from more scene-centric training images, we conduct pre-training with the COCO+ dataset and provide the corresponding transfer results in Table 2.7.

Visualizations of matches with in-box queries but low matching similarities. When formulating the semantic reweighting strategy, we assume that matches with in-box queries, which

Table 2.7: Transfer results from COCO+. The results of SlotCon and PixCon-SR are reported as the averages of 5, 3, 3, 5, and 3 independent runs for VOC detection, COCO detection and instance segmentation, Cityscapes segmentation, VOC segmentation, and ADE20k segmentation, respectively. Except for PixCon-SR, all the methods are region-level methods. (†: re-prod. w/official weights). © [2024] IEEE.

Method	**Dataset**	**VOC Detection**			**COCO**		**City. Seg.**	**VOC Seg.**	**ADE20k**
		AP	**AP$_{50}$**	**AP$_{75}$**	**AP^{bb}**	**AP^{mk}**	**mIoU**	**mIoU**	**mIoU**
ORL † [51]	COCO	55.8	82.1	62.3	40.2	36.4	75.4	70.7	-
UniVIP [15]		56.5	82.3	62.6	40.8	36.8	-	-	-
SlotCon † [14]		54.5	81.9	60.3	40.8	36.8	76.1	71.7	38.7
PixCon-SR (ours)		57.6	82.8	64.0	40.8	36.8	76.6	73.0	38.0
ORL [51]	COCO+	-	-	-	40.6	36.7	-	-	-
UniVIP [15]		58.2	83.3	**65.2**	41.1	37.1	-	-	-
SlotCon † [14]		57.0	83.0	63.4	**41.7**	**37.6**	76.6	**74.1**	**38.9**
PixCon-SR (ours)		**58.5**	**83.4**	**65.2**	41.2	37.1	**77.0**	73.9	38.8

lie in the intersected area of query and key views, are highly likely to own semantically consistent keys regardless of the query-key similarities, as they are guaranteed to have semantic correspondences in the key view. In Figure 2.6, we visualize the correspondences between in-box query pixels and their matched key pixels. We can observe that even in an early stage of training, most of the in-box queries with low matching similarities still have semantically consistent key pixels. This validates our assumption that in-box queries tend to have semantically consistent keys regardless of their matching similarities. As training goes further, the matches also get more accurate despite the magnitudes of similarities.

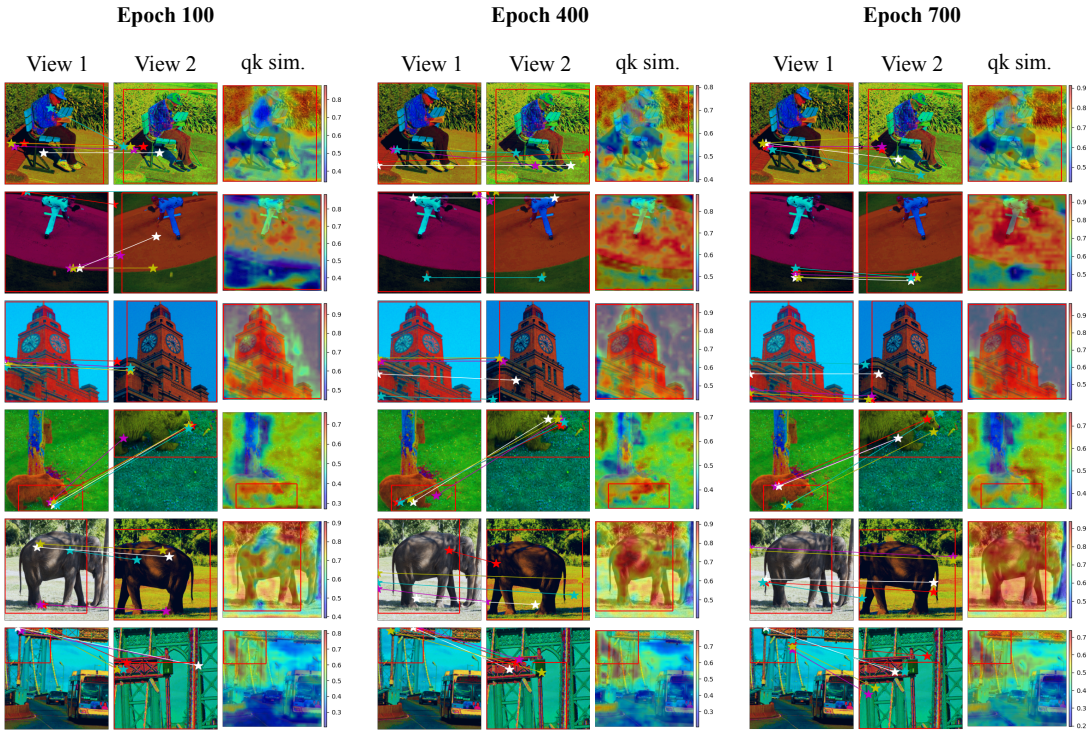

Figure 2.6: For each query view (view 1), we calculate the cosine similarities between its backbone features and those of the key view (view 2) at different training epochs. We keep five in-box query pixels that have the lowest similarities with their matched keys using similarity-based matching. The input images are randomly cropped, resized to 1024×1024, and then go through the other default data augmentations. The large input size is to more precisely visualize the correspondences. "qk sim." stands for the backbone feature similarities between the query and its matched key pixels and is only visualized for the query view. © [2024] IEEE.

2.6 Conclusion

In this paper, we exploited the potential of pixel-level learning on pre-training with scene images. We find that pixel-level learning baselines do not enjoy the same sophisticated training pipeline as employed in region-level methods. After training pipeline alignment, pixel-level methods can be improved to match the region-level methods' performance. Moreover, we show that pixel-level methods can also grasp regional semantics, where the key is the similarity-based positive matching strategy [1]. We eventually propose a semantic reweighting strategy to leverage both semantic and spatial cues to equip pixel-level learning with the capability of coping with semantically inconsistent scene image views. The semantic reweighting strategy helps pixel-level learning outperform or rival region-level methods, but with a much simpler methodology. We believe there is still under-explored potential for pixel-level learning, and we will keep exploring this direction in future work.

Chapter 3

Exploiting Contrastive Learning for Zero-Shot Video Summarization

3.1 Overview

In an era where video data are booming at an unprecedented pace, the importance of making the video browsing process more efficient has never been greater. Video summarization facilitates efficient browsing by creating a concise synopsis of the raw video, a topic that has been popular in research for many years. The rapid development of deep learning has significantly promoted the efficacy of video summarization tools [84]. Supervised approaches [85–88] leverage the temporal modeling power of LSTM (long short-term memory) [89] or self-attention mechanisms [90] and train them with annotated summaries. Heuristic training objectives such as diversity and representativeness have been applied using unsupervised methods [5, 91–96] to enforce a diverse selection of keyframes that are representative of the essential contents of videos.

Past unsupervised approaches have trained summarization models to produce diverse and representative summaries by optimizing feature similarity-based loss/reward functions. Many research works on visual representation learning have revealed that vision models pre-trained on

supervised or self-supervised tasks contain rich semantic signals, facilitating zero-shot transfer learning in tasks such as classification [9, 97], semantic segmentation [24], and object detection [98]. In this work, we propose leveraging the rich semantics encoded in pre-trained visual features to achieve zero-shot video summarization that outperforms previous heavily trained approaches and self-supervised pre-training to further enhance the zero-shot performance.

Specifically, we first define *local dissimilarity* and *global consistency* as two desirable criteria for localizing keyframe candidates. Inspired by the diversity objective, if a frame is distant from its nearest neighbors in the feature space, it encodes information that rarely appears in other frames. As a result, including such frames in the summary contributes to the diversity of its content. Such frames are considered to be decent key frame candidates as they enjoy high local dissimilarity, the naming of which leverages the definition of locality in the feature space in [99]. However, merely selecting frames based on dissimilarity may wrongly incorporate noisy frames that are not indicative of the video storyline. Therefore, we constrain the keyframes to be aligned with the video storyline by guaranteeing their high semantic similarity with the global cluster of the video frames, *i.e.* they are representative of (or globally consistent with) the video theme. Overall, the selected keyframes should enjoy a decent level of local dissimilarity to increase the content diversity in the summary and reflect the global video gist.

In contrast to previous works that required training to enforce the designed criteria, we directly quantify the proposed criteria into frame-level importance scores by utilizing contrastive losses for visual representation learning, *i.e.* alignment and uniformity losses [21]. The alignment loss calculates the distance between semantically similar samples, such as augmented versions of an input image, and minimizes this distance to ensure similarity between these positive samples in a contrastive learning setting. In our case, we directly apply the alignment loss to quantify the local dissimilarity metric. Uniformity loss is employed to regularize the overall distribution of features, with higher values indicating closely clustered features. This characteristic makes it well-suited for assessing the semantic consistency across a group of frames. To leverage this, we adapt the uniformity loss to evaluate the consistency between an individual frame and the entire set of video frames, which serves as a proxy for the global video storyline.

These two losses can then be utilized for *self-supervised contrastive refinement* of the features, where contrastive learning is applied to optimize feature distances, ultimately enhancing the accuracy of the calculated frame importance scores.

Nonetheless, background frames may feature dynamic content that changes frequently, making them distinct from even the most similar frames and resulting in local dissimilarity. At the same time, these frames might contain background elements that are common across a majority of the video frames, contributing to global consistency. For example, in a video of a car accident, street scenes are likely to appear consistently. Although these frames might differ due to moving objects, they remain generally consistent with most frames, on average, due to the shared background context. We propose mitigating the chances of selecting such frames by exploiting the observation that such background frames tend to appear in many different videos with diverse topics and, thus, are not unique to their associated videos, *e.g.*, street scenes in videos about car accidents, parades, city tours, etc. Specifically, we propose a *uniqueness filter* to quantify the uniqueness of frames, formulated by leveraging cross-video contrastive learning. An illustration of the difference between the proposed method and previous methods is provided in Figure 3.1.

Leveraging rich semantic information encoded in pre-trained visual features, we, for the first time, propose tackling training-free zero-shot video summarization and self-supervised pre-training to enhance the zero-shot transfer. Inspired by contrastive loss components [21], we achieve zero-shot summarization by quantifying frame importance into three metrics: local dissimilarity, global consistency, and uniqueness. The proposed method achieves better or competitive performance compared to previous methods while being training-free. Moreover, we introduce self-supervised contrastive refinement using unlabeled videos from YouTube-8M [100] to refine the feature distribution, which aids in training the proposed uniqueness filter and further enhances performance. This chapter is based on the conference and journal papers of this project [101, 102].

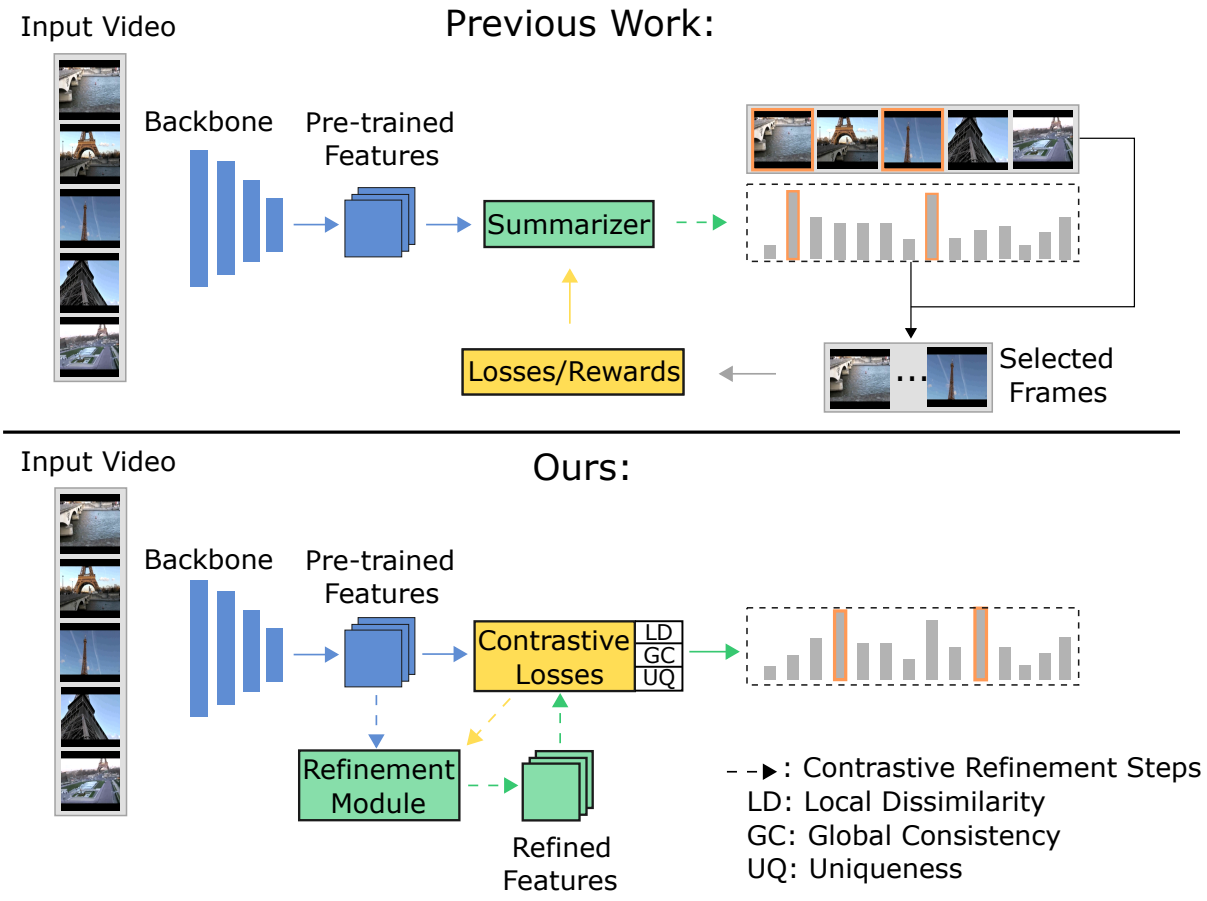

Figure 3.1: A comparison between our method and previous work. © [2023] IEEE.

3.2 Related Work

Early applications in video summarization focus on sports videos [103–105] for event detection and highlight video compilation. Later on, video summarization was explored in other domains such as instructional videos [4, 106–108], movies [109, 110], and general user videos [3]. Thanks to the excellent generalization capabilities of deep neural networks/features, the focus of video summarization research has been diverted to developing general-purpose summarization models for a diverse range of video domains.

As an initial step toward deep learning-based supervised video summarization, Zhang et al. [85] utilized a long short-term memory (LSTM) for modeling temporal information when

trained with human-annotated summaries, which sparked a series of subsequent works based on LSTM [86, 111–114]. The rise of Transformer [90] inspired a suite of methods leveraging self-attention mechanisms for video summarization [87,88,93,115–119]. Some works have explored spatiotemporal information by jointly using RNNs and convolutional neural networks (CNNs) [120–122] or used graph convolution networks [123, 124]. Video summarization leveraging multi-modal signals has also performed impressively [125–127].

Deep learning-based unsupervised methods mainly exploit two heuristics: diversity and representativeness. For diversity, some works [91, 92, 94, 124] have utilized a diversity loss derived from a repelling regularizer [128], guaranteeing dissimilarities between selected keyframes. It has also been formulated as a reward function optimized via policy gradient methods, as seen in [5, 129, 130]. Similarly, representativeness can be guaranteed by reconstruction loss [91, 93–95, 131] or reconstruction-based reward functions [5, 129, 130].

Unlike previous works, we tackle training-free zero-shot video summarization and propose a pre-training strategy for better zero-shot transfer. Specifically, we directly calculate frame importance by leveraging contrastive loss terms formulated in [21] to quantify diversity and representativeness. With features from a vision backbone pre-trained on supervised image classification tasks [132] and without any further training, the proposed contrastive loss-based criteria can already well-capture the frame contribution to the diversity and representativeness of the summary. The proposed self-supervised contrastive refinement can further boost the performance and leverage unlabeled videos for zero-shot transfer to test videos.

3.3 Preliminaries

Given the centrality of contrastive learning to our approach, we first introduce the relevant preliminaries, with a focus on instance discrimination as outlined in [48].

3.3.1 Instance Discrimination via the InfoNCE Loss

Contrastive learning [133] has become a cornerstone of self-supervised image representation

learning; throughout the years, it has received more attention from researchers. This method has been continuously refined to produce representations with exceptional transferability [6, 11, 21, 48, 99, 131, 134, 135]. Formally, given a set of N images $\mathcal{D} = \{I_n\}_{n=1}^{N}$, contrastive representation learning aims to learn an encoder f_θ such that the resulting features $f_\theta(I_n)$ can be readily leveraged by downstream vision tasks. A theoretically founded loss function with favorable empirical behaviors is InfoNCE loss [11]:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{I \in \mathcal{D}} -\log \frac{e^{f_\theta(I) \cdot f_\theta(I')/\tau}}{\sum_{J \in \mathcal{D}'(I)} e^{f_\theta(I) \cdot f_\theta(J)/\tau}}, \tag{3.1}$$

where I' is a positive sample for I, usually obtained through data augmentation, and $\mathcal{D}'(I)$ includes I' as well as all negative samples, *e.g.*, any other images. The operator "$\cdot$" is the inner product and τ is a temperature parameter. Therefore, the loss aims to pull the features of an instance closer to those of its augmented views while repelling them from the features of other instances, thus performing instance discrimination.

3.3.2 Contrastive Learning via Alignment and Uniformity

When normalized onto the unit hypersphere, the features learned through contrastive learning that yield strong downstream performance exhibit two notable properties. First, semantically related features tend to cluster closely on the sphere, regardless of specific details. Second, the overall information of the features is largely preserved, resulting in a joint distribution that approximates a uniform distribution [11, 134, 135]. Wang et al. [21] termed these two properties as *alignment* and *uniformity*.

The alignment metric computes the distance between the positive pairs [21]:

$$\mathcal{L}_{\text{align}}(\theta, \alpha) = \mathop{\mathbb{E}}_{(I, I') \sim p_{\text{pos}}} [\|f_\theta(I) - f_\theta(I')\|_2^\alpha], \tag{3.2}$$

where $\alpha > 0$, and p_{pos} is the distribution of positive pairs. The uniformity is defined as the average pairwise Gaussian potential between the overall features, as follows:

$$\mathcal{L}_{\text{uniform}}(\theta, \beta) = \log \left(\mathop{\mathbb{E}}_{I, J \overset{\text{i.i.d}}{\sim} p_{\text{data}}} [e^{-\beta \|f_\theta(I) - f_\theta(J)\|_2^2}] \right). \tag{3.3}$$

Here, p_{data} is typically approximated by the empirical data distribution, and β is commonly set to 2, as recommended by [21]. This metric promotes the overall feature distribution on the unit hypersphere to approximate a uniform distribution and can also directly quantify the uniformity of feature distributions [22]. Additionally, Equation (3.3) approximates the logarithm of the denominator in Equation (3.1) when the number of negative samples approaches infinity [21]. As demonstrated in [21], jointly minimizing Equations (3.2) and (3.3) leads to better alignment and uniformity of the features, meaning they become locally clustered and globally uniform [22].

In this paper, we employ Equation (3.2) to calculate the distance or dissimilarity between semantically similar video frame features, which helps measure frame importance based on local dissimilarity. We then apply a modified version of Equation (3.3) to assess the proximity between a specific frame and the overall information of the corresponding video, thereby estimating their semantic consistency. Additionally, by leveraging these two loss functions, we learn a nonlinear projection of the pre-trained features to enhance the local alignment and global uniformity of the projected features.

3.4 Proposed Method

We first define two metrics to quantify frame importance by leveraging rich semantic information in pre-trained features: local dissimilarity and global consistency. To guarantee that the metrics encode the diversity and representativeness of the summary, we conduct self-supervised contrastive refinement of the features, where an extra metric called uniqueness is defined to further strengthen the keyframes' quality. We provide a conceptual illustration of our approach in Figure 3.2.

3.4.1 Local Dissimilarity

Inspired by the diversity objective, we consider frames likely to result in a diverse summary as those conveying diverse information, even when compared to their nearest neighbors. Formally,

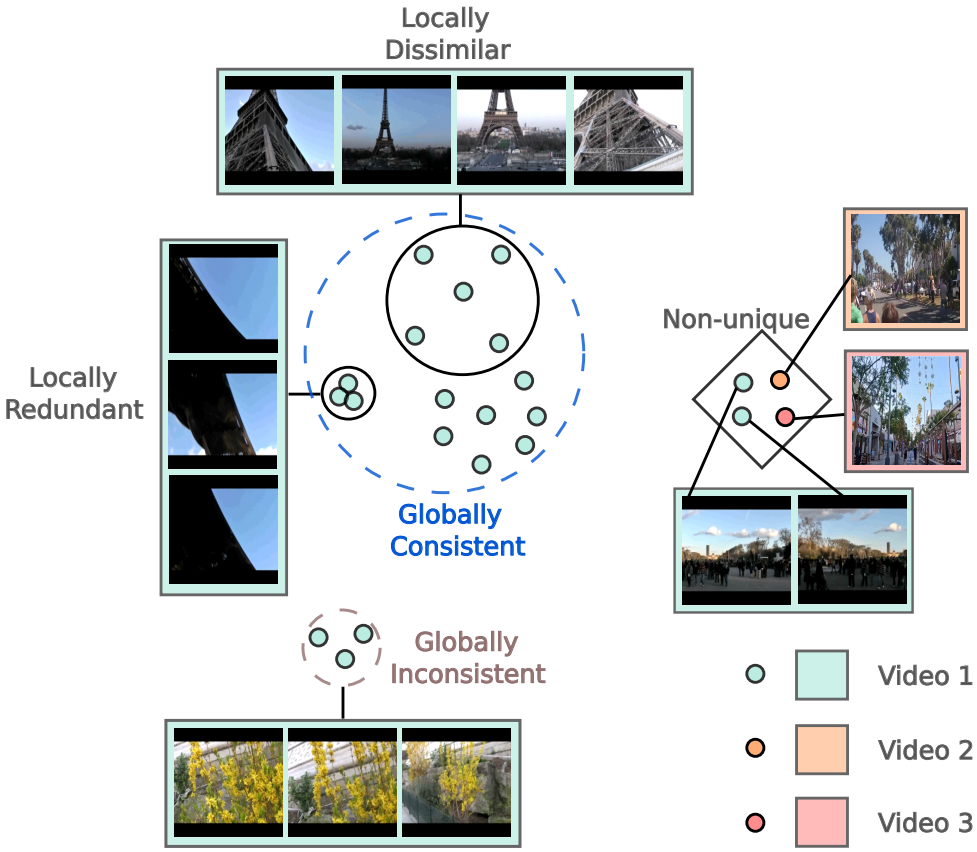

Figure 3.2: A conceptual illustration for the three metrics: local dissimilarity, global consistency, and uniqueness in the semantic space. The images come from the SumMe [3] and TVSum [4] datasets. The dots with the same color indicate features from the same video. For a concise demonstration, we only show one frame for "Video 2" and "Video 3" to show the idea of uniqueness. © [2023] IEEE.

given a video $\mathbf{V}$, we first extract deep features using the ImageNet [136] pre-trained vision backbone, *e.g.*, GoogleNet [132], denoted as F, such that $F(\mathbf{V}) = \{\mathbf{x}_t\}_{t=1}^{T}$, where $\mathbf{x}_t$ represents the deep feature for the t-th frame in $\mathbf{V}$, and T is the total number of frames in $\mathbf{V}$. Each feature is L2-normalized such that $\|\mathbf{x}_t\|_2 = 1$.

To define local dissimilarity for frames in $\mathbf{V}$, we first use cosine similarity to retrieve for each frame $\mathbf{x}_t$ a set $\mathcal{N}_t$ of top $K = aT$ neighbors, where a is a hyperparameter and K is rounded to the nearest integer. The local dissimilarity metric for $\mathbf{x}_t$ is an empirical approximation of

Equation (3.2), defined as the local alignment loss:

$$\mathcal{L}_{\text{align}}(\mathbf{x}_t) = \frac{1}{|\mathcal{N}_t|} \sum_{\mathbf{x} \in \mathcal{N}_t} \|\mathbf{x}_t - \mathbf{x}\|_2^2, \tag{3.4}$$

which measures the distance/dissimilarity between $\mathbf{x}_t$ and its semantic neighbors.

The larger the value of $\mathcal{L}_{\text{align}}(\mathbf{x}t)$, the more dissimilar $\mathbf{x}t$ is from its neighbors. Therefore, if a frame exhibits a certain distance from even its closest neighbors in the semantic space, the frames within its local neighborhood are likely to contain diverse information, making them strong candidates for keyframes. Consequently, $\mathcal{L}_{\text{align}}(\mathbf{x}_t)$ can be directly utilized as the importance score for $\mathbf{x}_t$ after appropriate scaling.

3.4.2 Global Consistency

$\mathcal{N}_t$ may contain semantically irrelevant frames if $\mathbf{x}_t$ has very few meaningful semantic neighbors in the video. Therefore, merely using Equation (3.4) for frame-wise importance scores is insufficient. Inspired by the reconstruction-based representativeness objective [91], we define another metric,, called global consistency, to quantify how consistent a frame is with the video gist by a modified uniformity loss based on Equation (3.3):

$$\mathcal{L}_{\text{uniform}}(\mathbf{x}_t) = \log \left(\frac{1}{T-1} \sum_{\substack{\mathbf{x} \neq \mathbf{x}_t, \\ \mathbf{x} \in F(\mathbf{V})}} e^{-2\|\mathbf{x}_t - \mathbf{x}\|_2^2} \right), \tag{3.5}$$

$\mathcal{L}_{\text{uniform}}(\mathbf{x}_t)$ measures the proximity between $\mathbf{x}_t$ and the remaining frames, bearing similarity to the reconstruction- and K-medoid-based objectives in [5, 91]. However, it obviates the need to train an autoencoder [91] or a policy network [5] by directly leveraging rich semantics in pre-trained features.

3.4.3 Contrastive Refinement

Equations (3.4) and (3.5) are computed using deep features pre-trained on image classification tasks, which may not inherently exhibit the local alignment and global uniformity described

in Section 3.3.2. To address similar challenges, Hamilton et al. [24] proposed contrastively refining self-supervised vision transformer features [9] for unsupervised semantic segmentation. They achieve this by freezing the feature extractor (to improve efficiency) and training only a lightweight projector. Following this approach, we also avoid fine-tuning the heavy feature extractor, in our case, GoogleNet, and instead train only a lightweight projection head.

Formally, given features $F(\mathbf{V})$ from the frozen backbone for a video, we feed them to a learnable module to obtain $\mathbf{z}_t = G_\theta(\mathbf{x}_t)$, where $\mathbf{z}_t$ is L2-normalized (we leave out the L2-normalization operator for notation simplicity). The nearest neighbors in $\mathcal{N}_t$ for each frame are still determined using the pre-trained features $\{\mathbf{x}_t\}_{t=1}^T$. Similar to [1, 99], we also observe collapsed training when directly using the learnable features for nearest neighbor retrieval, so we stick to using the frozen features.

With the learnable features, the alignment loss (local dissimilarity) and uniformity loss (global consistency) become (we slightly abuse the notation of $\mathcal{L}$ to represent losses both before and after transformation by G_θ):

$$\mathcal{L}_{\text{align}}(\mathbf{z}_t;\theta) = \frac{1}{|\mathcal{N}_t|}\sum_{\mathbf{z}\in\mathcal{N}_t}\|\mathbf{z}_t-\mathbf{z}\|_2^2, \tag{3.6}$$

$$\mathcal{L}_{\text{uniform}}(\mathbf{z}_t;\theta) = \log\left(\frac{1}{T-1}\sum_{\substack{\mathbf{z}\neq\mathbf{z}_t,\\ \mathbf{z}\in G_\theta(F(\mathbf{V}))}} e^{-2\|\mathbf{z}_t-\mathbf{z}\|_2^2}\right), \tag{3.7}$$

The joint loss function is as follows:

$$\mathcal{L}(\mathbf{z}_t;\theta) = \mathcal{L}_{\text{align}}(\mathbf{z}_t;\theta) + \lambda_1\mathcal{L}_{\text{uniform}}(\mathbf{z}_t;\theta), \tag{3.8}$$

where λ_1 is a hyperparameter balancing the two loss terms.

During the contrastive refinement, $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ will mutually resist each other for frames that have semantically meaningful nearest neighbors and are consistent with the video gist. Specifically, when a nontrivial number of frames beyond $\mathcal{N}_t$ also share similar semantic information with the anchor $\mathbf{z}_t$, these frames function as "hard negatives" that prevent $\mathcal{L}_{\text{align}}$ to be easily minimized [22, 99]. Therefore, only frames with moderate local dissimilarity and

global consistency will have balanced values for the two losses. In contrast, the other frames tend to have extreme values compared to those before the refinement.

3.4.4 The Uniqueness Filter

The two metrics defined above fail to account for the fact that locally dissimilar yet globally consistent frames can often be background frames with complex content that is related to most of the frames in the video. For example, dynamic cityscapes might frequently appear in videos recorded in urban settings.

To address this, we propose filtering out such frames by leveraging a common characteristic: they tend to appear in many different videos that do not necessarily share a common theme or context. For instance, city views might be present in videos about car accidents, city tours, or parades, while scenes featuring people moving around can appear across various contexts. Consequently, these frames are not unique to their respective videos. This concept has been similarly explored in weakly-supervised action localization research [137–139], where a single class prototype vector is used to capture all background frames. However, our approach aims to identify background frames in an unsupervised manner. Additionally, rather than relying on a single prototype, which can be too restrictive [140], we treat each frame as a potential background prototype. By identifying frames that are highly activated across random videos, we develop a metric to determine the “ background-ness ” of a frame.

To design a filter for eliminating such frames, we introduce an extra loss to Equation (3.8) that taps into cross-video samples. For computational efficiency, we aggregate the frame features in a video $\mathbf{V}_k$ with T_k frames into segments of equal length m. The learnable features, $\mathbf{z}$, in each segment, are average-pooled and L2-normalized to obtain segment features $\mathcal{S}_k = \{\mathbf{s}_l\}_{l=1}^{|\mathcal{S}_k|}$ with $|\mathcal{S}_k| = T_k/m$. To measure the proximity of a frame with frames from a randomly sampled batch of videos $\mathcal{B}$ (represented as segment features), including $\mathcal{S}_k$, we again leverage Equa-

tion (3.3) to define the uniqueness loss for $\mathbf{z}_t \in \mathbf{V}_k$ as follows:

$$\mathcal{L}_{\text{unique}}(\mathbf{z}_t;\theta) = \log\left(\frac{1}{A}\sum_{\mathcal{S}\in\mathcal{B}/\mathcal{S}_k}\sum_{\mathbf{s}\in\mathcal{S}} e^{-2\|\mathbf{z}_t-\mathbf{s}\|_2^2}\right), \tag{3.9}$$

where $A = \sum_{\mathcal{S}\in\mathcal{B}/\mathcal{S}_k}|\mathcal{S}|$ is the normalization factor. A large value of $\mathcal{L}_{\text{unique}}$ means that $\mathbf{z}_t$ has nontrivial similarity with segments from randomly gathered videos, indicating that it is likely to be a background frame. When jointly optimized with Equations (3.8) and (3.9) the process will be easy to minimize for unique frames, for which most elements of $\mathbf{s}$ are semantically irrelevant and can be safely repelled. It is not the case for the background frames with semantically similar $\mathbf{s}$, as the local alignment loss keeps strengthening the closeness of semantically similar features.

As computing Equation (3.9) requires random videos, it is not as straightforward to convert Equation (3.9) to importance scores after training. To address this, we train a model $H_{\hat{\theta}}$ whose last layer is a sigmoid unit to mimic $1-\bar{\mathcal{L}}_{\text{unique}}(\mathbf{z}_t;\theta)$, where $\bar{\mathcal{L}}_{\text{unique}}(\mathbf{z}_t;\theta)$ is $\mathcal{L}_{\text{unique}}(\mathbf{z}_t;\theta)$ scaled to $[0,1]$ over t. Denoting $y_t = 1-\text{sg}(\bar{\mathcal{L}}_{\text{unique}}(\mathbf{z}_t;\theta))$ and $r_t = H_{\hat{\theta}}(\text{sg}(\mathbf{z}_t))$, where "sg" stands for stop gradients, we define the loss for training the model as follows:

$$\mathcal{L}_{\text{filter}}(\mathbf{z}_t;\hat{\theta}) = -y_t\log r_t + (1-y_t)\log(1-r_t). \tag{3.10}$$

3.4.5 The Full Loss and Importance Scores

With all the components, the loss for each frame in a video is as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{z}_t;\theta,\hat{\theta}) = \mathcal{L}_{\text{align}}(\mathbf{z}_t;\theta) + \lambda_1\mathcal{L}_{\text{uniform}}(\mathbf{z}_t;\theta)\\ +\lambda_2\mathcal{L}_{\text{unique}}(\mathbf{z}_t;\theta) + \lambda_3\mathcal{L}_{\text{filter}}(\mathbf{z}_t;\hat{\theta}),\end{aligned} \tag{3.11}$$

where we fix both λ_2 and λ_3 as 0.1 and only tune λ_1.

Scaling the local dissimilarity, global consistency, and uniqueness scores to $[0,1]$ over t, the frame-level importance score is defined as follows:

$$p_t = \bar{\mathcal{L}}_{\text{align}}(\mathbf{z}_t;\theta)\bar{\mathcal{L}}_{\text{uniform}}(\mathbf{z}_t;\theta)\bar{H}_{\hat{\theta}}(\mathbf{z}_t) + \epsilon, \tag{3.12}$$

which ensures that the importance scores are high only when all three terms have significant magnitude. The parameter ϵ is included to prevent zero values in the importance scores, which

helps stabilize the knapsack algorithm used to generate the final summaries. Since these scores are derived from three independent metrics, they may lack the temporal smoothness typically provided by methods like RNNs [85] or attention networks [88]. To address this, we apply Gaussian smoothing to the scores within each video, aligning our method with previous work that emphasizes the importance of temporal smoothness in score generation.

3.5 Experiments

3.5.1 Datasets and Settings

Datasets. In line with previous studies, we evaluate our method on two benchmarks: TVSum [4] and SumMe [3]. TVSum consists of 50 YouTube videos, each annotated by 20 individuals who provide importance scores for every two-second shot. SumMe includes 25 videos, each with 15 to 18 reference binary summaries. Following the protocol established by [85], we use the OVP (50 videos) and YouTube (39 videos) datasets [141] to augment both TVSum and SumMe. Additionally, to assess whether our self-supervised approach can benefit from a larger video dataset, we randomly selected approximately 10,000 videos from the YouTube-8M dataset [100], which contains 3,862 video classes with highly diverse content.

Evaluation Setting. Following prior work, we evaluate our model's performance using five-fold cross-validation, where the dataset (either TVSum or SumMe) is randomly divided into five splits. The reported results are the average across these five splits. In the canonical setting (C), training is performed only on the original splits of the two evaluation datasets. In the augmented setting (A), we expand the training set in each fold with three additional datasets (*e.g.*, SumMe, YouTube, and OVP when evaluating on TVSum). In the transfer setting (T), all videos from TVSum (or SumMe) are reserved for testing, while the other three datasets are used for training. Additionally, we introduce a new transfer setting where training is exclusively conducted on the collected YouTube-8M videos, and evaluation is performed on TVSum or SumMe. This setting is intended to assess whether our model can benefit from a larger volume of data.

3.5.2 Evaluation Metrics

F1 score. Denoting A as the set of frames in a ground-truth summary and B as the set of frames in the corresponding generated summary, we can calculate precision and recall as follows:

$$\text{Precision} = \frac{|A \cap B|}{|A|}, \ \text{Recall} = \frac{|A \cap B|}{|B|}, \tag{3.13}$$

with which we can calculate the F1 score by the following:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.14}$$

We follow [85] to deal with multiple ground-truth summaries and to convert importance scores into summaries.

Rank correlation coefficients. Recently, Otani et al. [142] highlighted that F1 scores can be unreliable and may yield relatively high values even for randomly generated summaries. To address this issue, they proposed using rank correlation coefficients, specifically Kendall's τ [143] and Spearman's ρ [144], to evaluate the correlation between predicted and ground-truth importance scores. For each video, we first compute the coefficient value between the predicted importance scores and the scores provided by each annotator, then average these values across all annotators for that video. The final results are obtained by averaging the correlation coefficients across all videos.

3.5.3 Summary Generation

We follow previous work to convert importance scores to key shots. Specifically, we use the KTS algorithm [145] to segment videos into temporally consecutive shots and then average the importance scores within each shot to compute the shot-level importance scores. The final key shots are chosen to maximize the total score while guaranteeing that the summary length does not surpass 15% of the video length. The maximization is conducted by solving the knapsack problem based on dynamic programming [4]. Otani et al. [142] pointed out that using average frame importance scores as shot-level scores will drastically increase the F1 score for the TV-Sum dataset, and they recommended using the sum of scores to alleviate the problem. However,

F1 scores reported by previous works mostly rely on averaging importance scores for shot-level scores. We also report our F1 scores in the same way as they did but focus on analyzing the rank correlation values for comparison and analysis.

3.5.4 Implementation Details

We follow prior studies by using GoogleNet [132] pre-trained features as the default for standard experiments. For experiments involving YouTube-8M videos, we utilize the quantized Inception-V3 [146] features provided by the dataset [100]. Both types of features are pre-trained on ImageNet [136]. The contrastive refinement module appended to the feature backbone is a lightweight Transformer encoder [90], and so is the uniqueness filter.

Following [92], we standardized each video to have an equal length by using random sub-sampling for longer videos and nearest-neighbor interpolation for shorter videos. Similar to [92], we did not observe much difference when using different lengths, and we fixed the frame count at 200.

The model appended to the feature backbone for contrastive refinement is a stack of Transformer encoders with multi-head attention modules [90]. There are two training scenarios: 1) Training with TVSum [4], SumMe [3], YouTube, and OVP [141], divided into the canonical, augmented, and transfer settings; 2. Training with a subset of videos from the YouTube-8M dataset [100]. We refer to the training in the first scenario as *standard* and the second as *YT8M*. The pre-trained features are first projected into 128 dimensions for training in both scenarios using a learnable, fully connected layer. The projected features are then fed into the Transformer encoders. The model architecture and associated optimization details are outlined in Table 3.1. Training the 10,000 YouTube-8M videos takes approximately 6 minutes for 40 epochs on a single NVIDIA RTX A6000.

We tune two hyperparameters: The ratio a, which determines the size of the nearest neighbor set $\mathcal{N}_t$ and the coefficient λ_1, which controls the balance between the alignment and uniformity losses.

Table 3.1: Model and optimization details. © [2023] IEEE.

	Layers	Heads	d_{model}	d_{head}	d_{inner}	Optimizer	LR	Weight Decay	Epoch
Standard	4	1	128	64	512	Adam	0.0001	0.0001	40
YT8M	4	8	128	64	512	Adam	0.0001	0.0005	40

3.5.5 Quantitative Results

In this section, we compare our results with previous work and conduct the ablation study for different components of our method.

Training-free zero-shot performance. As shown in Tables 3.2 and 3.3, $\bar{\mathcal{L}}^*_{\text{align}}$ and $\bar{\mathcal{L}}^*_{\text{uniform}}$ directly computed using GoogleNet [132] pre-trained features, achieve performance superior to most methods in terms of τ, ρ, and F1 score. Notably, the correlation coefficients τ and ρ surpass supervised methods, *e.g.*, (0.1345, 0.1776) v.s. dppLSTM's (0.0298, 0.0385) and SumGraph's (0.094, 0.138) for TVSum. Although DR-DSN$_{2000}$ has slightly better performance in terms of τ and ρ for TVSum, it has to reach the performance after 2000 epochs of training, while our results are directly obtained with simple computations using the same pre-trained features as those also used by DR-DSN.

More training videos are needed for the contrastive refinement. For the results in Tables 3.2 and 3.3, the maximum number of training videos is only 159, coming from the SumMe augmented setting. For the canonical setting, the training set size is 40 videos for TVSum and 20 for SumMe. Without experiencing many videos, the model tends to overfit specific videos and cannot generalize well. This is similar to the observation in contrastive representation learning, where a larger amount of data, whether from a larger dataset or obtained through data augmentation, helps the model generalize better [6, 9]. Therefore, the contrastive refinement results in Tables 3.2 and 3.3 hardly outperform those computed using pre-trained features.

Contrastive refinement on YouTube-8M videos and transfer to TVSum. The model generalizes to the test videos better when sufficient training videos are given, as shown by the

Table 3.2: Ablation results in terms of τ and ρ, along with their comparisons to previous work in the canonical setting. DR-DSN$_{60}$ refers to the DR-DSN trained for 60 epochs; similarly, DR-DSN$_{2000}$. Our scores with superscript $*$ are directly computed from pre-trained features. The results were generated with $(\lambda_1, a) = (0.5, 0.1)$. **Bold** scores = best among supervised; blue = best without annotations; † = vision-language methods. © [2023] IEEE.

	TVSum		**SumMe**	
Method	τ	ρ	τ	ρ
Human baseline [147]	0.1755	0.2019	0.1796	0.1863
Supervised				
VASNet [88, 147]	0.1690	0.2221	0.0224	0.0255
dppLSTM [85, 142]	0.0298	0.0385	−0.0256	−0.0311
SumGraph [124]	0.094	0.138	–	–
Multi-ranker [147]	**0.1758**	**0.2301**	0.0108	0.0137
Clip-It† [126]	0.108	0.147	–	–
A2Summ† [127]	0.137	0.165	**0.108**	**0.129**
Unsupervised				
DR-DSN$_{60}$ [5, 142]	0.0169	0.0227	0.0433	0.0501
DR-DSN$_{2000}$ [5, 147]	0.1516	0.1980	−0.0159	−0.0218
SUM-FCN$_{\text{unsup}}$ [92, 147]	0.0107	0.0142	0.0080	0.0096
SUM-GAN [91, 147]	−0.0535	−0.0701	−0.0095	−0.0122
CSNet + GL + RPE [96]	0.070	0.091	–	–
Training-free				
$\bar{\mathcal{L}}^*_{\text{align}}$	0.1055	0.1389	0.0960	0.1173
$\bar{\mathcal{L}}^*_{\text{align}}$ & $\bar{\mathcal{L}}^*_{\text{uniform}}$	0.1345	0.1776	0.0819	0.1001
Contrastively refined				
$\bar{\mathcal{L}}_{\text{align}}$	0.1002	0.1321	0.0942	0.1151
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$	0.1231	0.1625	0.0689	0.0842
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$	0.1388	0.1827	0.0585	0.0715
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ & $\bar{H}_{\hat{\theta}}$	0.1609	0.2118	0.0358	0.0437

Table 3.3: Ablation results regarding F1 and their comparisons with previous unsupervised methods. The **boldfaced** results are the best ones. Please refer to Table 3.2's caption for the notation and text for analysis. © [2023] IEEE.

	TVSum			**SumMe**		
Method	C	A	T	C	A	T
Unsupervised						
DR-DSN$_{60}$ [5]	57.6	58.4	57.8	41.4	42.8	42.4
SUM-FCN$_{\text{unsup}}$ [92]	52.7	–	–	41.5	–	39.5
SUM-GAN [91]	51.7	59.5	–	39.1	43.4	–
UnpairedVSN [94]	55.6	–	55.7	47.5	–	41.6
CSNet [95]	58.8	59	59.2	**51.3**	**52.1**	45.1
CSNet + GL + RPE [96]	59.1	–	–	50.2	–	–
SumGraph$_{\text{unsup}}$ [124]	59.3	**61.2**	57.6	49.8	**52.1**	**47**
Training-free						
$\bar{\mathcal{L}}^{*}_{\text{align}}$	56.4	56.4	54.6	43.5	43.5	39.4
$\bar{\mathcal{L}}^{*}_{\text{align}}$ & $\bar{\mathcal{L}}^{*}_{\text{uniform}}$	58.4	58.4	56.8	47.2	46.07	41.7
Contrastively refined						
$\bar{\mathcal{L}}_{\text{align}}$	54.6	55.1	53	46.8	47.1	41.5
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$	58.8	59.9	57.4	46.7	48.4	41.1
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$	53.8	56	54.3	45.2	45	45.3
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ & $\bar{H}_{\hat{\theta}}$	**59.5**	59.9	**59.7**	46.8	45.5	43.9

results for TVSum in Table 3.4. After the contrastive refinement, the results with only $\bar{\mathcal{L}}^*_{\text{align}}$ are improved from (0.0595, 0.0779) to (0.0911, 0.1196) for τ and ρ. We can also observe improvement over $\bar{\mathcal{L}}^*_{\text{align}}$ & $\bar{\mathcal{L}}^*_{\text{uniform}}$ brought by contrastive refinement.

Contrastive refinement on YouTube-8M videos and transfer to SumMe. The reference summaries in SumMe are binary scores, and summary lengths are constrained to be within 15% of the video lengths. Therefore, the majority of the reference summary receives exactly zero scores. The contrastive refinement may still enhance the confidence scores for these regions, which receive zero scores from annotators due to the 15% constraint. This can ultimately reduce the average correlation with the reference summaries, as seen in Table 3.4.

Table 3.4: The transfer evaluation setting with the YouTube-8M dataset, where the training is solely conducted on the collected YouTube-8M videos and then evaluated on TVSum and SumMe. The results from DR-DSN [5] are also provided for comparison. © [2023] IEEE.

	TVSum			**SumMe**		
Method	F1	τ	ρ	F1	τ	ρ
Unsupervised						
DR-DSN [5]	51.6	0.0594	0.0788	39.8	-0.0142	-0.0176
Training-free						
$\bar{\mathcal{L}}^*_{\text{align}}$	55.9	0.0595	0.0779	45.5	0.1000	0.1237
$\bar{\mathcal{L}}^*_{\text{align}}$ & $\bar{\mathcal{L}}^*_{\text{uniform}}$	56.7	0.0680	0.0899	42.9	0.0531	0.0649
Contrastively refined						
$\bar{\mathcal{L}}_{\text{align}}$	56.2	0.0911	0.1196	46.6	0.0776	0.0960
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$	57.3	0.1130	0.1490	40.9	0.0153	0.0190
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$	58.1	0.1230	0.1612	48.7	0.0780	0.0964
$\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ & $\bar{H}_{\hat{\theta}}$	59.4	0.1563	0.2048	43.2	0.0449	0.0553

Suppose that the predicted scores are refined to have sufficiently high confidence for regions with nonzero annotated scores; in this case, they are likely to be selected by the knapsack algorithm used to compute the F1 scores. Therefore, we consider scores that achieve both

high F1 and high correlations to be of high quality, as the former tends to overlook the overall correlations between the predicted and annotated scores [142], while the latter focuses on their overall ranked correlations but places less emphasis on prediction confidence. This analysis may explain why the contrastive refinement for $\bar{\mathcal{L}}^*_{\text{align}}$ improves the F1 score but decreases the correlations.

The effect of $\bar{\mathcal{L}}_{\text{align}}$. As can be observed in Tables 3.2-3.4, solely using $\bar{\mathcal{L}}_{\text{align}}$ can already well-quantify the frame importance. This indicates that $\bar{\mathcal{L}}_{\text{align}}$ successfully selects frames with diverse semantic information, which are indeed essential for a desirable summary. Moreover, we assume that diverse frames form the foundation of a good summary, consistently using $\bar{\mathcal{L}}_{\text{align}}$ for further ablations.

The effect of $\bar{\mathcal{L}}_{\text{uniform}}$. $\bar{\mathcal{L}}_{\text{uniform}}$ measures how consistent a frame is with the context of the whole video, thus helping remove frames with diverse contents that are hardly related to the video theme. It is shown in Tables 3.2 and 3.4 that incorporating $\bar{\mathcal{L}}_{\text{uniform}}$ helps improve the quality of the frame importance for TVSum. We now discuss why $\bar{\mathcal{L}}_{\text{uniform}}$ hurts SumMe performance.

Compared to TVSum videos, many SumMe videos already contain consistent frames due to their slowly evolving properties. Such slowly evolving features can be visualized by T-SNE plots in Figure 3.3. For videos with such consistent content, the $\bar{\mathcal{L}}_{\text{uniform}}$ tends to be high for most of the frames. We show the normalized histogram of $\mathcal{L}^*_{\text{uniform}}$ for both TVSum and SumMe videos in Figure 3.4. As can be observed, SumMe videos have distinctly higher $\mathcal{L}^*_{\text{uniform}}$ than those of TVSum videos. Consequently, for videos possessing monotonous content, most of the frames share a similar visual cue, such as the background, and the frames that are most likely to be keyframes are those with abrupt or novel content. Therefore, the global consistency metric, $\bar{\mathcal{L}}^*_{\text{uniform}}$, is not discriminative enough to be sufficiently helpful and may alleviate the importance of frames with novel content. As a result, the other two metrics, local dissimilarity and uniqueness, are the main roles that determine keyframes in such videos, as shown in Table 3.2-3.4.

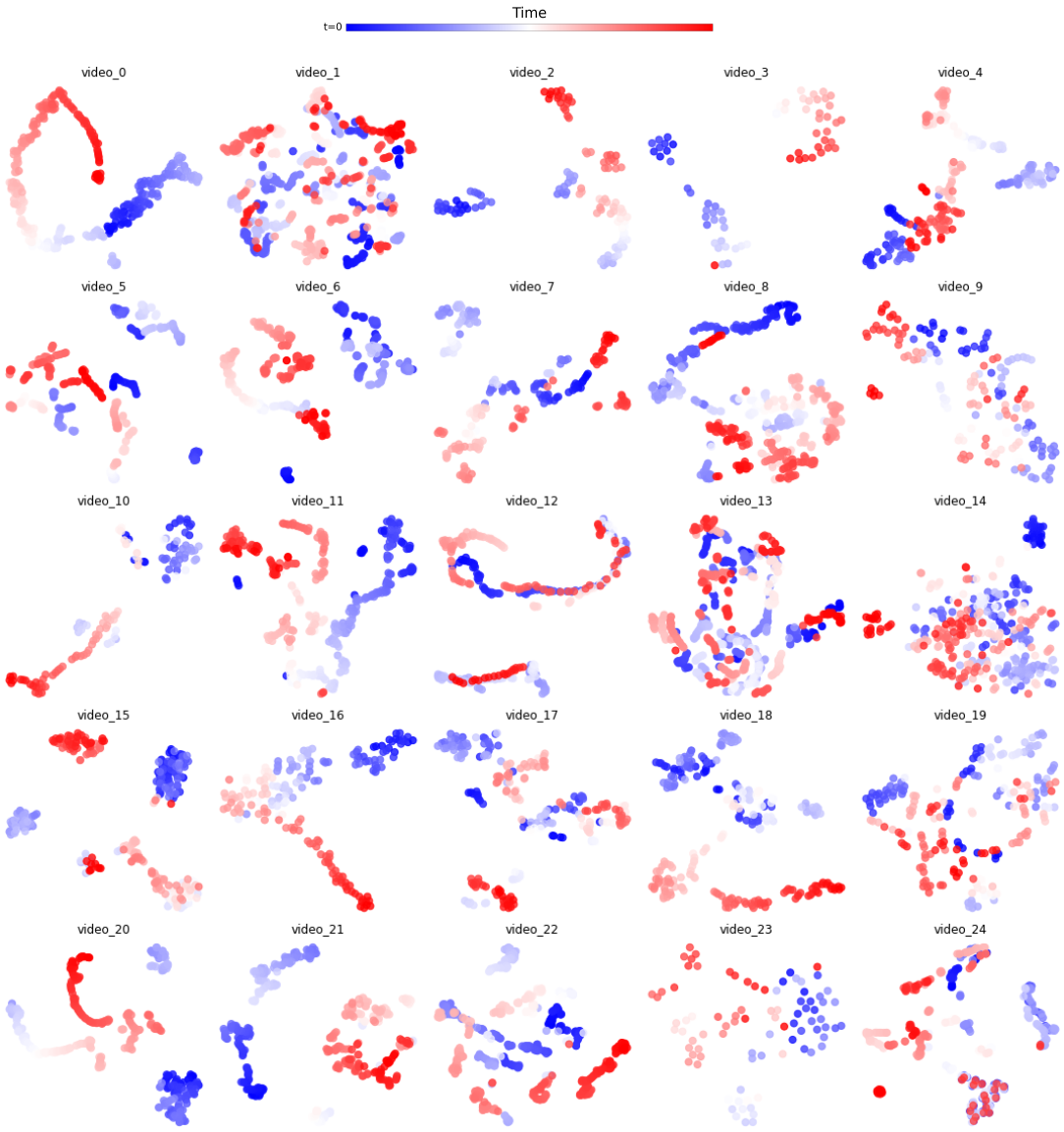

Figure 3.3: TSNE plots for all 25 SumMe videos. As can be observed, many videos contain features that slowly evolve and maintain an overall similarity among all the frames. © [2023] IEEE.

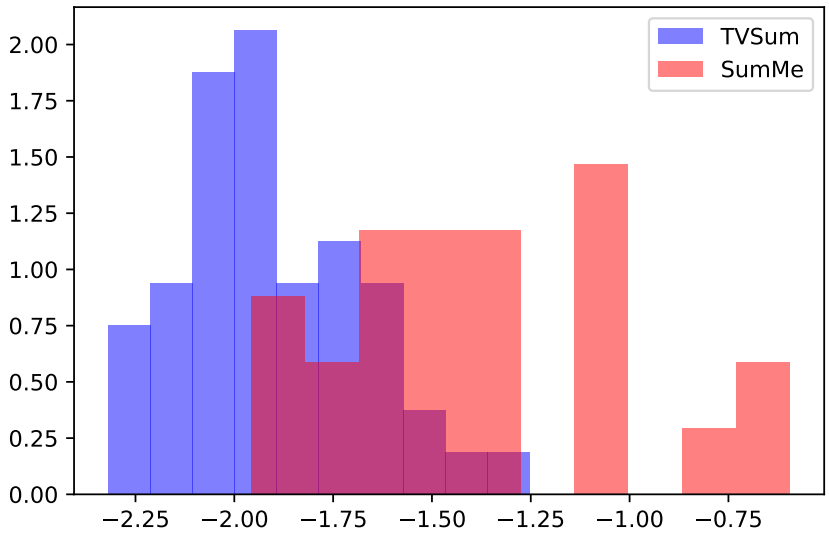

Figure 3.4: The histogram (density) of $\bar{\mathcal{L}}^{*}_{\text{uniform}}$ (before normalization) for TVSum and SumMe videos. SumMe videos have distinctly higher values than those for TVSum videos. © [2023] IEEE.

The effect of the uniqueness filter $\bar{H}_{\hat{\theta}}$. As shown in Tables 3.2 and 3.3, although $\bar{H}_{\hat{\theta}}$ works well for TVSum videos, it hardly brings any benefits to the SumMe videos. Thus, the good performance of the uniqueness filter for TVSum may be due to the relatively straightforward nature of the background frames in TVSum, which are easily identified by the uniqueness filter even with training on only a few videos. Therefore, we suppose that $\bar{H}_{\hat{\theta}}$ needs to be trained on more videos to filter out more challenging background frames such that it can generalize to a wider range of videos. This is validated by the $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$ results in Table 3.4, which indicate both decent F1 scores and correlation coefficients for both TVSum and SumMe. The TVSum performance can be further boosted when $\bar{\mathcal{L}}_{\text{uniform}}$ is incorporated.

Comparison with DR-DSN [5] on YouTube-8M. As per Table 3.2, DR-DSN is the only unsupervised method that matches our performance in terms of τ and ρ and has an official implementation available. We trained DR-DSN on our dataset of YouTube-8M videos to compare it against our method. As shown in Table 3.4, DR-DSN has difficulty generalizing to the

evaluation videos.

Ablations over λ_1 and a. As shown in Figure 3.5, when $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$ is used to produce importance scores, a larger a will make the TVSum performance unstable in terms of both F1 and correlation coefficients, although the SumMe performance is relatively more stable with respect to a. We hypothesize that when a becomes larger, the nearest neighbor set becomes noisier, diminishing the effectiveness of both the alignment loss during training and the local dissimilarity metric (post-training alignment loss) used for generating importance scores, due to the inclusion of semantically irrelevant neighbors. For λ_1, smaller values generally perform better when a has a reasonable value, as larger values of λ_1 tend to make the uniformity loss suppress the alignment loss. Similarly, too small λ_1 will make the alignment loss suppress the uniformity loss, as we observed unstable training when further decreasing λ_1. As shown in Figure 3.6, the analysis of the interaction between λ_1 and a when using $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ is used to produce importance scores, similar to that in Figure 3.5. However, we can see that the performance was improved for TVSum but undermined for SumMe due to incorporating $\bar{\mathcal{L}}_{\text{uniform}}$.

Ablation on model sizes. Table 3.5 shows the ablation results for different sizes of the Transformer encoder [90], where the number of layers and the number of attention heads are varied. Meanwhile, we compare the results with those obtained from DR-DSN [5] trained on the same collected YouTube-8M videos, as DR-DSN has the best τ and ρ among past unsupervised methods and is the only one that has a publicly available official implementation. As can be observed, the model performance is generally stable with respect to the model sizes, and we choose 4L8H. Moreover, the DR-DSN has difficulty generalizing well to the test videos when trained on the YouTube-8M videos.

Comparing the effects of different pre-trained features. As our method can directly compute importance scores using pre-trained features, it is also essential for it to be able to work with different kinds of pre-trained features. To prove this, we computed and evaluated the importance scores generated with 2D supervised features, 3D supervised features, and 2D self-supervised features in Table 3.6. Different 2D features, whether supervised or self-supervised,

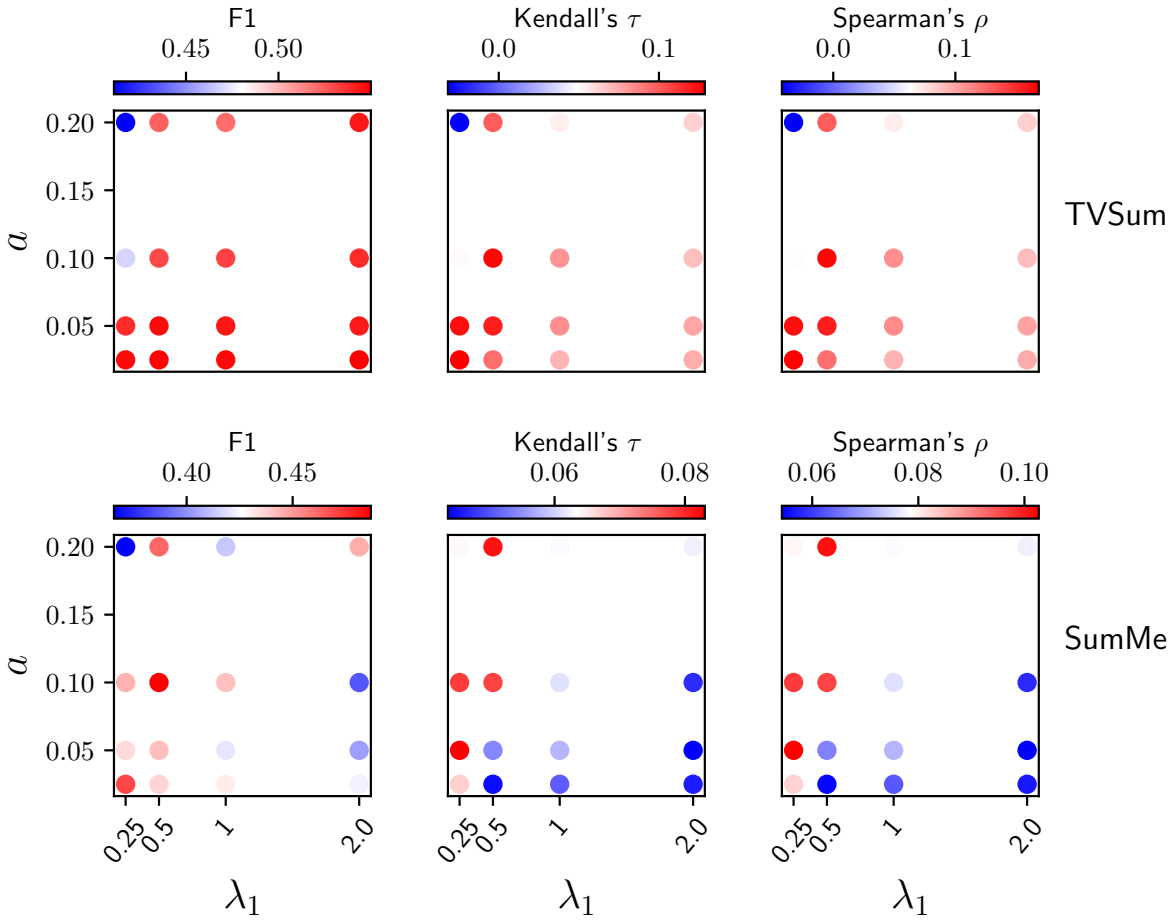

Figure 3.5: Ablation results over λ_1 and a when using $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$ to produce importance scores.

© [2023] IEEE.

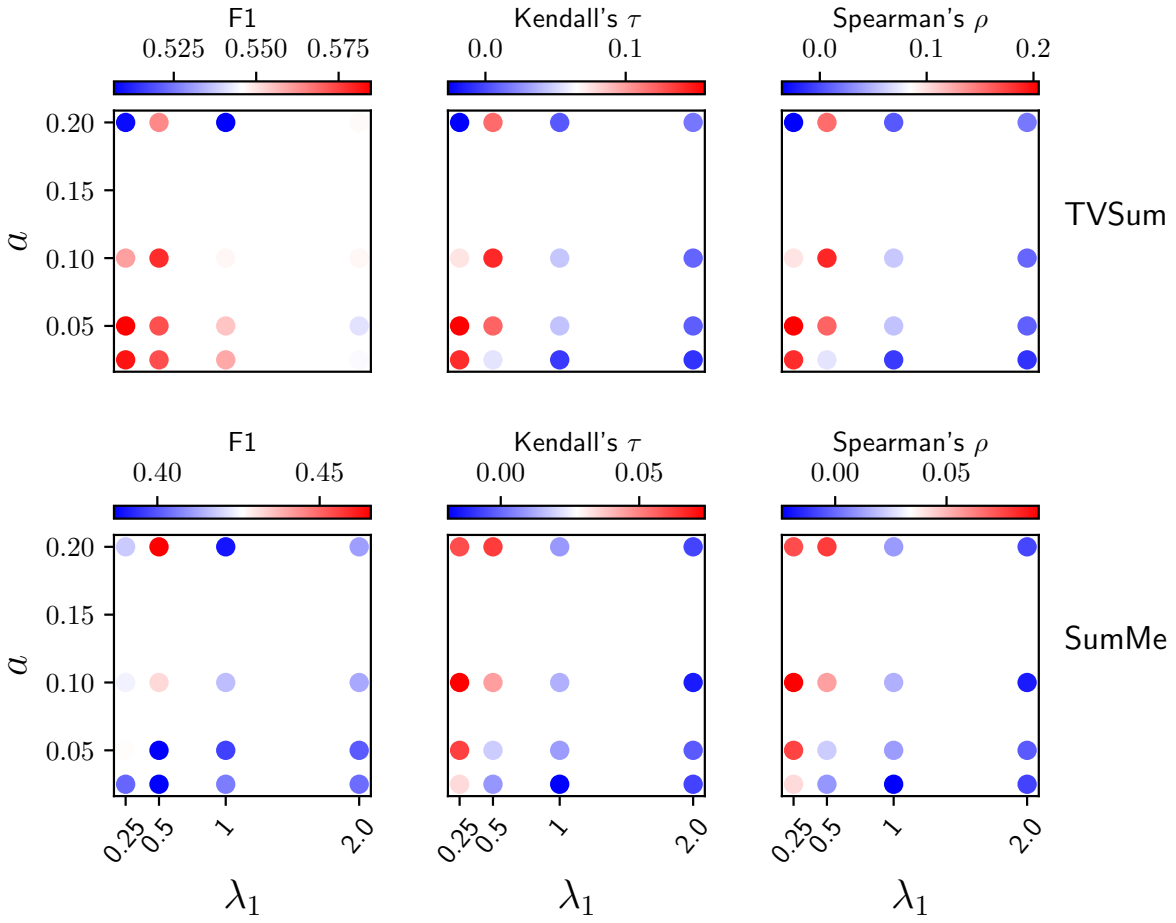

Figure 3.6: Ablation results over λ_1 and a when using $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ to produce importance scores. © [2023] IEEE.

Table 3.5: Ablation results for the model size and comparison with DR-DSN [5] trained on the same YouTube-8M videos, where 2L2H represents " 2 layers 2 heads " and similarly for the rest. All three components $\bar{\mathcal{L}}_{\text{align}}$, $\bar{H}_{\hat{\theta}}$ and $\bar{\mathcal{L}}_{\text{uniform}}$ are used with $(a, \lambda_1) = (0.05, 0.25)$ for both SumMe and TVSum for fair comparison with DR-DSN's representativeness-based training objective. © [2023] IEEE.

	TVSum			**SumMe**		
Method	F1	τ	ρ	F1	τ	ρ
DR-DSN [5]	51.6	0.0594	0.0788	39.8	-0.0142	-0.0176
2L2H	58.0	0.1492	0.1953	42.9	0.0689	0.0850
2L4H	58.1	0.1445	0.1894	42.8	0.0644	0.0794
2L8H	58.8	0.1535	0.2011	44.0	0.0584	0.0722
4L2H	57.4	0.1498	0.1963	45.3	0.0627	0.0776
4L4H	58.3	0.1534	0.2009	43.1	0.0640	0.0790
4L8H	58.5	0.1564	0.2050	42.7	0.0618	0.0765

all delivered decent results. Differences exist but are trivial. The conclusion that $\bar{\mathcal{L}}_{\text{unif}}$ helps TVSum but undermines SumMe also holds for most of the features. Based on this, we conclude that as long as the features contain decent semantic information learned from supervision or self-supervision, they are enough to efficiently compute the importance scores. The performance of these features transferred to different downstream image tasks does not necessarily generalize to our method for video summarization, as the latter only requires reliable semantic information (quantified as dot products) to calculate heuristic metrics for video frames.

Notably, our method does not perform optimally with 3D supervised video features. This outcome is anticipated because these 3D features are trained to encode information based on video-level labels, thus capturing less detailed semantic information in individual frames, which is crucial for our method. Still, such 3D features contain part of the holistic information of the associated video and may be a good vehicle for video summarization, which can benefit from such information.

Table 3.6: Evaluation results with different pre-trained features. The results were produced under the transfer setting with $a = 0.1$. © [2023] IEEE.

	TVSum						**SumMe**					
	$\bar{\mathcal{L}}^*_{\text{align}}$			$\bar{\mathcal{L}}^*_{\text{align}}$ & $\bar{\mathcal{L}}^*_{\text{unif}}$			$\bar{\mathcal{L}}^*_{\text{align}}$			$\bar{\mathcal{L}}^*_{\text{align}}$ & $\bar{\mathcal{L}}^*_{\text{unif}}$		
Method	F1	τ	ρ	F1	τ	ρ	F1	τ	ρ	F1	τ	ρ
Supervised (2D)												
VGG19 [148]	50.62	0.0745	0.0971	55.91	0.1119	0.1473	45.16	0.0929	0.1151	43.28	0.0899	0.1114
GoogleNet [132]	54.67	0.0985	0.1285	57.09	0.1296	0.1699	41.89	0.0832	0.1031	40.97	0.0750	0.0929
InceptionV3 [146]	55.02	0.1093	0.1434	55.63	0.0819	0.1082	42.71	0.0878	0.1087	42.30	0.0688	0.0851
ResNet50 [76]	51.19	0.0806	0.1051	55.19	0.1073	0.1410	42.30	0.0868	0.1076	43.86	0.0737	0.0914
ResNet101 [76]	51.75	0.0829	0.1081	54.88	0.1118	0.1469	42.32	0.0911	0.1130	44.39	0.0736	0.0913
ViT-S-16 [149]	53.48	0.0691	0.0903	56.15	0.1017	0.1332	40.30	0.0652	0.0808	40.88	0.0566	0.0701
ViT-B-16 [149]	52.85	0.0670	0.0873	56.15	0.0876	0.1152	42.10	0.0694	0.0860	41.65	0.0582	0.0723
Swin-S [150]	52.05	0.0825	0.1082	57.58	0.1120	0.1475	41.18	0.0880	0.1090	41.63	0.0825	0.1022
Supervised (3D)												
R3D50 [151]	52.09	0.0590	0.0766	53.35	0.0667	0.0869	37.40	0.0107	0.0138	41.03	0.0150	0.0190
R3D101 [151]	49.77	0.0561	0.0727	52.15	0.0644	0.0834	33.62	0.0173	0.0216	34.96	0.0212	0.0264
Self-supervised (2D)												
MoCo [7]	51.31	0.0797	0.1034	55.97	0.1062	0.1390	42.01	0.0768	0.0953	43.19	0.0711	0.0882
DINO-S-16 [9]	52.50	0.0970	0.1268	57.57	0.1200	0.1583	42.77	0.0848	0.1050	42.67	0.0737	0.0913
DINO-B-16 [9]	52.48	0.0893	0.1170	57.02	0.1147	0.1515	41.07	0.0861	0.1066	44.14	0.0679	0.0843
BEiT-B-16 [152]	49.64	0.1125	0.1468	56.34	0.1270	0.1665	36.91	0.0554	0.0686	38.48	0.0507	0.0629
MAE-B-16 [153]	50.40	0.0686	0.0892	54.58	0.1013	0.1327	40.32	0.0560	0.0695	39.46	0.0484	0.0601

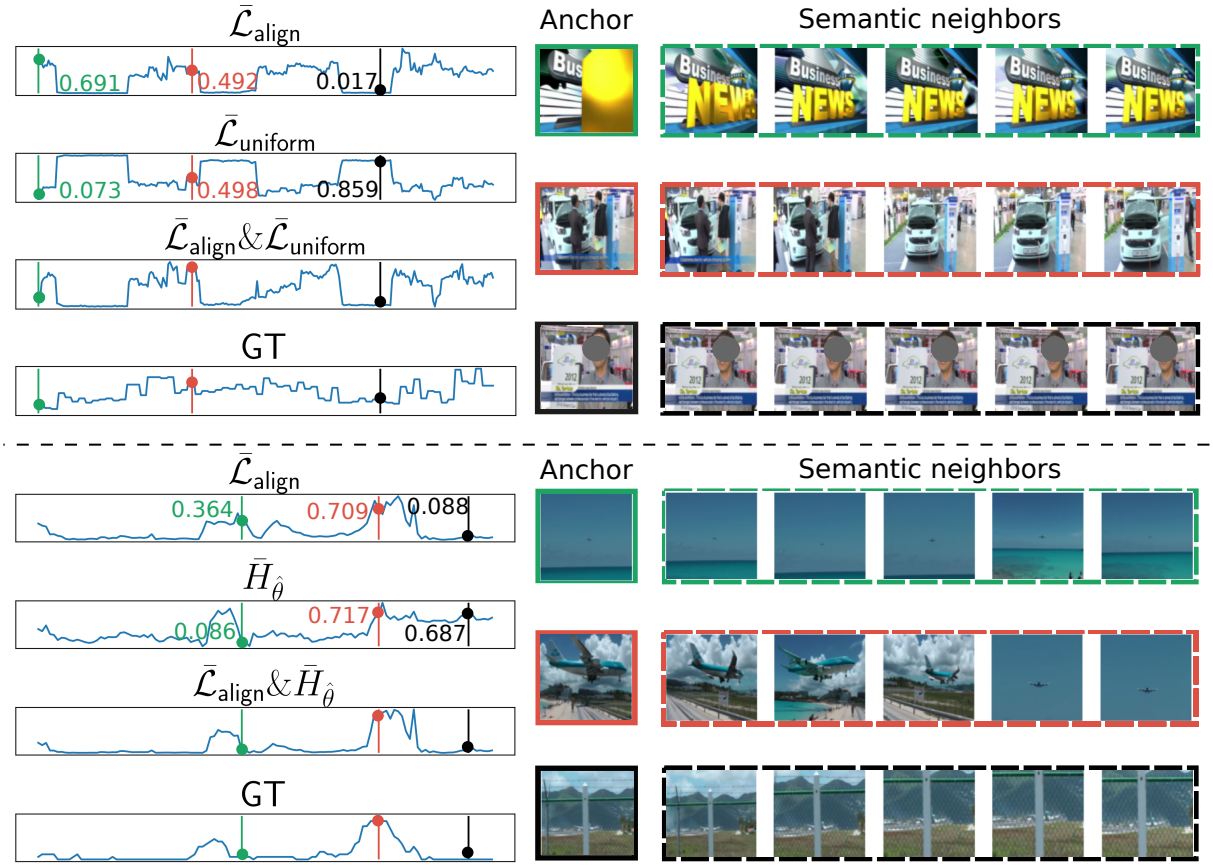

Figure 3.7: The qualitative analysis of two video examples. The left column contains importance scores, where "GT" stands for ground truth. The green bar selects an anchor frame with high $\bar{\mathcal{L}}_{\text{align}}$ but low $\bar{\mathcal{L}}_{\text{uniform}}$ or $\bar{H}_{\hat{\theta}}$, the red bar selects one with non-trial magnitude for both metrics, and the black bar selects one with low $\bar{\mathcal{L}}_{\text{align}}$ but high $\bar{\mathcal{L}}_{\text{uniform}}$ or $\bar{H}_{\hat{\theta}}$. We show five samples from the top 10 semantic nearest neighbors within the dashed boxes on the right for each selected anchor frame. © [2023] IEEE.

3.5.6 Qualitative Results

We show the effect of the local dissimilarity ($\bar{\mathcal{L}}_{\text{align}}$), the global consistency ($\bar{\mathcal{L}}_{\text{uniform}}$), and the uniqueness scores generated by the uniqueness filter $\bar{H}_{\hat{\theta}}$ in Figure 3.7. We visualize and discuss the effects in pairs, *i.e.* $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{\mathcal{L}}_{\text{uniform}}$ and $\bar{\mathcal{L}}_{\text{align}}$ & $\bar{H}_{\hat{\theta}}$. In the upper half of Figure 3.7, the green bar selects a frame with high local similarity but low global consistency, which is a title frame with a disparate appearance and hardly conveys any valuable information about the video. While the black bar selects a frame related to the main content of the video (an interview), it has semantic neighbors with almost the same look and is less likely to contain diverse semantics. The red bar selects a frame with moderate local dissimilarity and global consistency. This frame, along with its semantic neighbors, conveys diverse information; for example, the car with or without people surrounding it. Moreover, it is highly relevant to the overall video context: an interview at a car company.

For the lower half of Figure 3.7, the green bar selects a frame with information noticeably different from its neighbors, *e.g.*, the sea occupies different proportions of the scene. However, such a frame can appear in any video with water scenes, rendering it not unique to the belonging video. Hence, its uniqueness score is low. The black bar selects a frame with an object specifically belonging to this video in the center, but the local semantic neighborhood around it hardly conveys diverse information. The red bar selects a frame with both high local dissimilarity and high uniqueness, which is the frame related to the gist of the video: St. Maarten landing.

3.6 Conclusion

We make the first attempt to approach training-free, zero-shot video summarization by leveraging pre-trained deep features. We utilize contrastive learning to propose three metrics, local dissimilarity, global consistency, and uniqueness, to generate frame importance scores. The proposed metrics directly enable the creation of summaries with quality that is better or competitive compared to previous supervised or unsupervised methods requiring extensive training. Moreover, we propose contrastive pre-training on unlabeled videos to further boost the quality

of the proposed metrics, the effectiveness of which has been verified by extensive experiments. It would be interesting to explore multi-modal zero-hot video summarization for future work.

Chapter 4

Video Large Language Models Can Summarize to Localize

4.1 Overview

Video tasks that involve event-level and time-sensitive reasoning, such as temporal action localization [154–157], dense captioning [158–161], and grounded video question answering [162–165], require models to comprehend the content of videos and precisely identify when specific events occur by outputting event segment timestamps. While specialized models excel at individual tasks, they often struggle to generalize across different tasks, especially those involving complex reasoning. Recently, advancements in Video Large Language Models (Video LLMs) [166–172] have opened new avenues for unifying these tasks within a single framework by leveraging their powerful vision-and-language understanding and generation capabilities [171–175].

However, Video LLMs face significant challenges when it comes to temporal localization, particularly in generating precise timestamps of localized video segments. Initial efforts enable these models to output timestamps by representing them with numeric language tokens [171, 172] or augmenting the LLM's vocabulary with specialized timestamp tokens

[161, 172, 176]. Unfortunately, LLMs struggle with numerical data and often produce inconsistent or inaccurate results when handling numbers [177, 178]. Moreover, introducing new tokens necessitates extensive pre-training data and computational resources to adapt the models effectively [161, 172, 176]. Subsequent works have attempted various strategies to improve LLMs' ability to handle timestamps, such as formatting numeric timestamps to the same length to alleviate LLM's burden in precisely capturing them [173], designing complex fusion mechanisms between visual and textual tokens [171, 173, 179, 180], or learning specialized embeddings to represent event boundaries [174]. Despite these efforts, accurately and efficiently generating timestamps remains a significant challenge for video LLMs in temporal grounding tasks.

While current efforts to enable video LLMs to generate timestamps have incrementally improved their temporal grounding performance, we approach the problem differently by making the LLM's output entirely *timestamp-free*. Specifically, we point out that the timestamps are essentially a summary of the localized video segments, expressed in a format that facilitates extracting the segments from the original video. This perspective highlights another crucial pitfall in enforcing video LLMs to output timestamps: it requires a significant leap from dense and language-aligned visual tokens to abstract and uninformative timestamp tokens. For instance, the LLM needs to first capture the visual tokens relevant to the input query, determine what the boundary visual tokens are, map such visual tokens to their associated timestamp tokens, and finally output such timestamps. Such a one-shot strategy usually poses substantial challenges to exploiting LLMs' semantic reasoning capability [181–183].

Inspired by the widely adopted Chain-of-Thought [181, 183] (CoT) prompting technique that utilizes the model's reasoning path as a bridge from the input to the final answer, we propose to let video LLMs produce a textual summary of the query-related segment, which serves as the bridge to the final timestamp outputs. For example, when the query is a short event tag or an action label, the output textual summaries consist of context-rich descriptions for relevant video segments. For complex temporal reasoning tasks such as grounded video question answering, such textual summaries can also take the form of a CoT reasoning path, which analytically navigates the content of relevant segments to help generate the final answer. Given

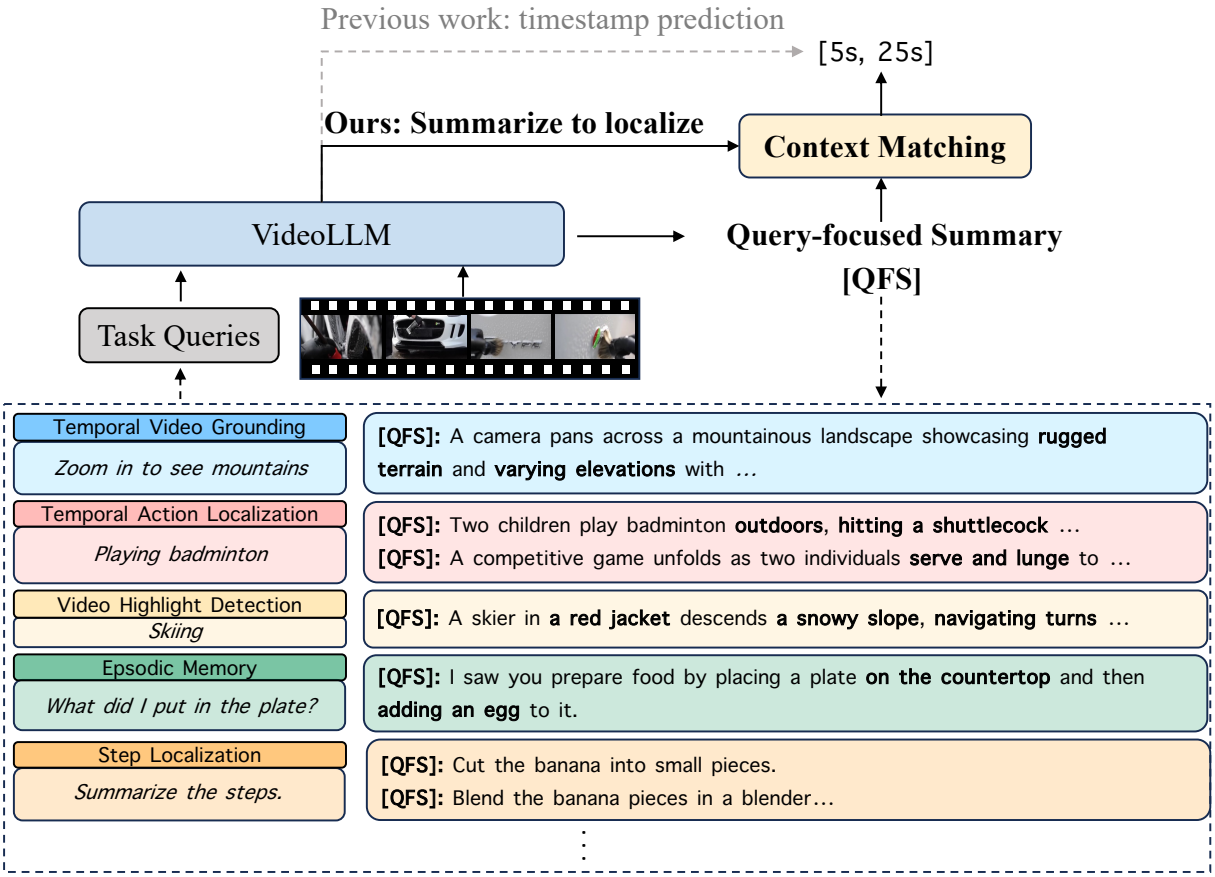

Figure 4.1: The proposed **S2L** framework features two components: (1) the **Query-Focused Summarization** task that requires the LLM to generate query-focused summaries of the video based on the input user query, and (2) the **Context Matching** module optimized by contrastive learning, designed to ground the semantic information encoded in the query-focused summaries back to the video frames, thus achieving temporal localization purposes. Compared to previous works that focus on generating uninformative and semantically poor timestamps, S2L emphasizes the use of the powerful semantic understanding of the LLM and the integration of generative and discriminative learning.

the textual summary, we utilize a simple yet effective *context matching* mechanism driven by contrastive learning to extract the timestamps from the video segments based on the contextual information shared between the textual summaries and the visual input. The proposed pipeline also effectively removes the timestamp generation part for tasks requiring both timestamps and segment-wise captions, such as dense video captioning [158, 161] and step localization [184], by directly decoding the timestamps from the segment-wise captions/summaries via the context matching mechanism.

As a result, we unify a suite of event-level time-sensitive video tasks with a **S**ummarize-to-**L**ocalize framework, coined **S2L**, by fully exploiting the LLMs' intrinsic semantic understanding and retrieval capability and obviating the use of timestamps. An illustration of the proposed S2L framework is shown in Figure 4.1. To facilitate such a framework, we contribute an instruction tuning dataset, **ETSum**, which focuses on equipping the model with the capability of handling **E**vent-level **T**ime-sensitive reasoning by **Sum**marization based on a timestamp-centric dataset ETInstruct [174]. The proposed framework outperforms previous Video LLM-based temporal localization approaches across grounding, dense video captioning, and complex reasoning tasks.

4.2 Related Work

Video Large Language Models. Early efforts to enable LLMs to perform video-level tasks involved using LLMs as agents that process video clip-level captions and, through chain-of-thought reasoning and tool use, execute corresponding tasks [185–188]. While these agents have shown promising results, they are limited by the performance of specialist models used as tools. The advent of end-to-end multimodal pretraining [10, 46, 189] and instruction fine-tuning [190] has led to a suite of powerful video LLMs [166–172]. Recent studies have demonstrated that these models excel in temporal reasoning over very long videos, benefiting from the long-context processing abilities of LLMs [191–193] and dynamic visual token compression techniques [170, 194, 195]. However, they do not consider event-level video tasks that

require temporal localization. To address this, some works have proposed to fine-tune pre-trained video LLMs with temporal grounding data to explicitly output timestamps of the localized segments [171, 173, 175, 176, 179]. Nevertheless, such timestamp-based strategies have encountered various issues, including training difficulties, unsatisfactory performance, and increased computational overhead. In this work, we show that relying solely on language outputs is more effective for video LLMs to handle temporal localization tasks and aligns better with video LLMs' intrinsic multi-modal reasoning capabilities.

Event-Level and Time-Sensitive Video Tasks. Video tasks such as moment retrieval [42], highlight detection [42, 196, 197], video synopsis generation [4, 197], action localization [154–157], and dense video captioning [158,186], invariably involve localizing salient event segments in a video given a user-specified query, where oftentimes the precise timestamps of such events are needed. Tasks like dense video captioning and grounded video question answering [162–165] also involve captioning and complex reasoning regarding localized events. Traditionally, such tasks have been approached by specialist models with task-specific designs, trained on data from their respective domains. Efforts have also been made to develop unified specialist models for different localization-only tasks [198, 199].

Recently, the development of time-sensitive video LLMs has enabled the unification of both localization and generation tasks. Models like TimeChat [171] and VTimeLLM [175] fine-tune pre-trained video LLMs [167] to perform temporal localization by outputting numeric timestamp tokens. While these approaches demonstrate the models' capabilities in such tasks, they achieve less satisfactory performance. Some methods [161,172] augment the LLM's vocabulary with a set of learnable timestamp tokens, which require large-scale pre-training to be effective. Subsequent works focus on improving the compatibility between LLMs and timestamp outputs by unifying the lengths of numeric tokens with long padding [173] and/or fusing numeric tokens with input visual/textual tokens via interleaving or learnable fusion modules [176, 179], inevitably introducing computational overhead. ETChat [174] proposed fine-tuning the model to estimate the embeddings of event boundary frames via a single newly introduced token, but neglected the rich contexts within the event segment itself.

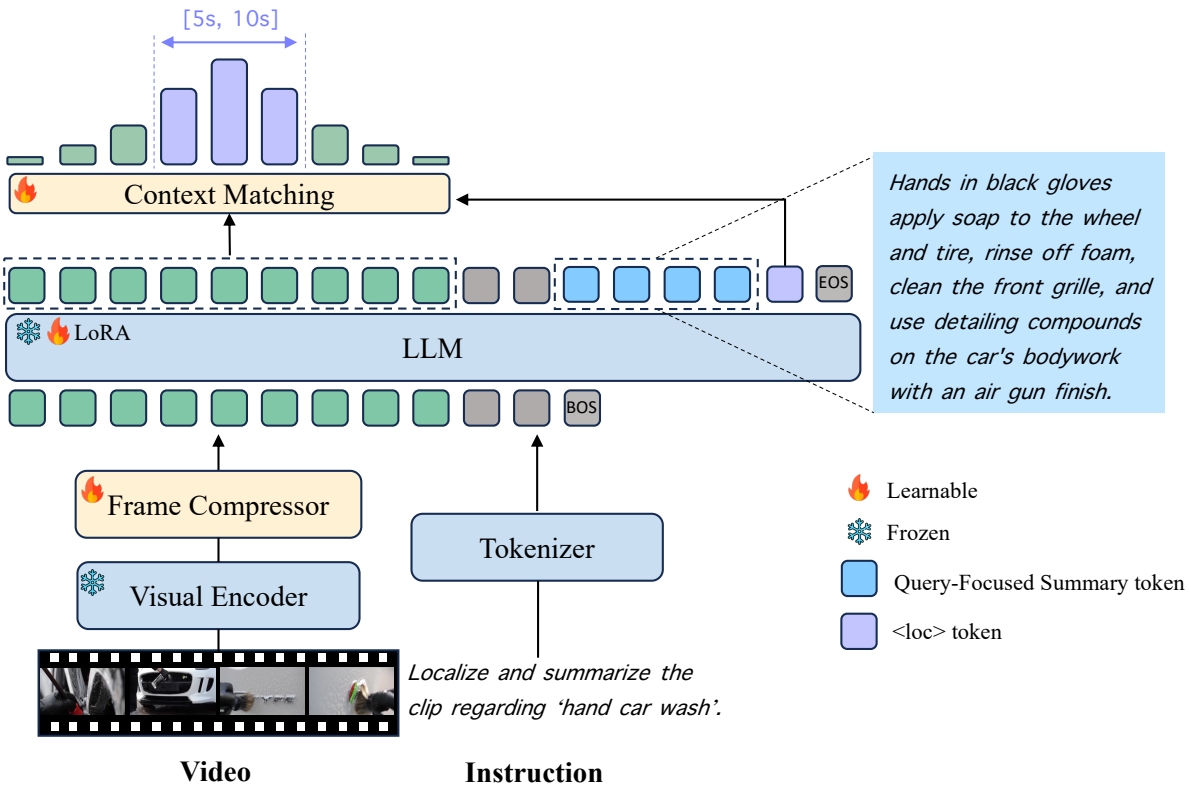

Figure 4.2: The architecture of the proposed S2L framework.

In contrast, we consider that the timestamps are merely summaries of the localized segments that are expressed in a highly abstract but convenient format for retrieving the segments. Past works have shown that LLMs have difficulties dealing with highly abstract outputs in both text-only [181–183] and multi-modal scenarios [200, 201]. Therefore, we propose to replace the uninformative and highly abstract timestamps with LLM-friendly and context-rich textual summaries as the VideoLLM's output. Producing the timestamps of the localized segments then becomes a by-product of matching the shared contextual information encoded in visual and textual embeddings. We show that this unified, semantically rich output format is more effective and generalizable than previous timestamp-based approaches.

4.3 Method

In this section, we first present the definition of event-level and time-sensitive video tasks and then introduce our S2L framework by describing the architecture of the Video LLM and the proposed context matching module. We then present the training and inference procedures of the framework and conclude by explaining the creation process of the instruction fine-tuning dataset, which is used to train the VideoLLM to perform temporal localization by query-focused summarization.

4.3.1 Task Definition

Event-level and time-sensitive (ET) video tasks require understanding and explicitly identifying the temporal locations of the events of interest. Previous studies tend to conduct evaluations on different sub-tasks and datasets, which causes difficulties in comparing their pros and cons. Recently, the ETBench benchmark [174] has been proposed to unify a suite of ET video tasks for comprehensive evaluation. It consists of four major tasks: referring, grounding, dense captioning, and complex understanding. Each of which contains a set of sub-tasks with different fine-grained requirements.

Given a video and an ET video task instruction, the outputs required from the model can be text-only $\mathbf{X}_{\text{text}}$ (referring), timestamp-only $\mathbf{X}_{\text{time}} = \{(t_m^s, t_m^e)\}_{m=1}^{M}$ (grounding), where t_m^s and t_m^e are the start and end timestamps of the m-th localized segment and M is the total number of the localized segments. Tasks involving captioning or reasoning, such as dense video captioning and grounded question answering, require both $\mathbf{X}_{\text{text}}$ and $\mathbf{X}_{\text{time}}$. Specialist models usually consist of two separate modules for $\mathbf{X}_{\text{text}}$ and $\mathbf{X}_{\text{time}}$, respectively. Recent time-sensitive video LLMs utilize the numeric tokens or learnable time tokens to represent timestamps in the LLMs' input and output space such that the same LLM can output both $\mathbf{X}_{\text{text}}$ and $\mathbf{X}_{\text{time}}$.

Due to the LLMs' limitations in dealing with tokenized timestamps [174], the proposed S2L framework lets the LLM focus on outputting $\mathbf{X}_{\text{text}}$ and uses a separate lightweight module to decode the timestamps $\mathbf{X}_{\text{time}}$ from the LLM's output. We introduce the different components of

S2L in the following sections.

4.3.2 Model Architecture

The overall architecture of S2L is presented in Figure 4.2. Given a video $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$, a pre-trained visual encoder extracts for each of the T frames a set of patch features $\mathbf{P} \in \mathbb{R}^{K\times C}$, where K is the number of patches and C is the feature dimension. Following [174], the frame-level patch features are sent into a frame compressor, which uses a Q-Former [46] as a resampler, an attention-based adaptive pooling module and a linear projection layer to compress the patch features into a single feature vector $\mathbf{e}_v^t \in \mathbb{R}^{1\times C}$. The frame compressor is applied to all frames to generate a set of frame-level features $\mathbf{E}_v = \{\mathbf{e}_v^t\}_{t=1}^T$. The instructions that contain task queries are tokenized and encoded into $\mathbf{E}_q = \{\mathbf{e}_q^l\}_{l=1}^L$, where $\mathbf{e}_q^l \in \mathbb{R}^{1\times C}$ and L is the number of instruction tokens. Finally, the frame and text features are concatenated and sent into the LLM for response generation.

Previous methods usually add a timestamp injection step when preparing the LLM inputs, fusing the timestamp tokens with the visual and/or textual inputs to build the connection between each frame and its associated timestamp for the LLM to more effectively generate $\mathbf{X}_{\text{time}}$. However, as it has been shown that LLMs struggle to handle different forms of timestamp representations [173, 174, 176] and have intrinsic limitations in coping with highly abstract outputs [181, 201], we propose to obviate the use of timestamps with the LLM and directly instruct the model to localize the segments related to the query by summarizing all the relevant contextual cues into language outputs. Therefore, the output will be the textual summary $\mathbf{X}_{\text{text}}$ alone. For referring tasks where the referred timestamps are needed in the input, we follow [174] to use a `<vid>` token in the input to represent them. The model is then trained to generate the summary with the language modeling loss:

$$\mathcal{L}_{\text{LM}} = -\sum_{n=1}^{N} \log P(\mathbf{x}_n|\mathbf{E}_v, \mathbf{E}_q, \mathbf{X}_{\text{text},<n}), \tag{4.1}$$

where N is the number of tokens in the output summary, $\mathbf{X}_{\text{text}} = \{\mathbf{x}_n\}_{n=1}^N$, and $\mathbf{X}_{\text{text},<n} = \{\mathbf{x}_{n'}\}_{n'=1}^n$.

Though ETChat proposes to fine-tune the LLM to approximate the event boundary frame embedding encoded in a newly added <vid> token, it only focuses on boundary information while overlooking the holistic context in each event. In contrast, the output summary from S2L is context-rich and can facilitate more accurate localization of the event.

4.3.3 Context Matching Module

To obtain the precise timestamps of interested video segments without requiring the LLM to output them, we propose a context matching module that produces a set of context matching scores between the output summary tokens and the video frames based on the shared contextual semantics encoded in their LLM hidden states. However, LLM hidden states are learned to facilitate generation tasks and may not exist in a space where the semantic information is discriminative enough for the localization task [174]. Therefore, we project the hidden states of the visual input and the output summary into another space using two learnable projection modules F^{vis} and F^{sum} to obtain the visual projection features $\mathbf{H}^{\text{vis}} \in \mathbb{R}^{T\times C}$ and the summary projection features for the m-th localized segment $\mathbf{H}^{\text{sum}}_m \in \mathbb{R}^{N\times C}$, respectively.

To better capture from the output summary the contexts essential for localization, we choose to introduce a <loc> token into the LLM's vocabulary, and force the LLM to end the summary of each segment with it. The <loc> token thus functions as a separator between the summaries of different segments, and its hidden features are a compact attention-based aggregation of the previous summary tokens' hidden features. As a result, the last element in $\mathbf{H}^{\text{sum}}_m$ can be extracted as $\mathbf{h}^{\texttt{<loc>}}_m \in \mathbb{R}^C$. The context matching scores are computed as the cosine similarities between $\mathbf{h}^{\texttt{<loc>}}_m$ and $\mathbf{H}^{\text{vis}}$:

$$\mathbf{S}_m = \text{cos\_sim}(\mathbf{H}^{\text{vis}}, \mathbf{h}^{\texttt{<loc>}}_m) \in \mathbb{R}^T \tag{4.2}$$

To obtain more discriminative context matching scores, we optimize a contrastive context matching loss. Given the ground-truth temporal intervals $\{(t^s_m, t^e_m)\}_{m=1}^M$, the context matching

loss is formulated as:

$$\mathcal{L}_{\text{CM}} = \frac{1}{M}\sum_{m=1}^{M} l_m, \tag{4.3}$$

$$l_m = -\frac{1}{t_m^e - t_m^s}\sum_{t=t_m^s}^{t_m^e} \log \frac{\exp(\mathbf{S}_{m,t}/\tau)}{\sum_{t'=1}^{T}\exp(\mathbf{S}_{n,t'}/\tau)}, \tag{4.4}$$

$\mathcal{L}_{\text{CM}}$ essentially encourages the context matching scores to be high between the summary projection features and those of the query-related frames, from which the LLM is supposed to collect the contextual cues for summary generation. Different from previous specialist temporal localization models, the context matching module does not involve intricate designs and many training parameters, as it is built upon a Video LLM that provides rich multi-modal features. We will show that this simple design can achieve competitive performance on the ET video tasks.

4.3.4 Training and Inference

To adapt a VideoLLM to the S2L framework, we use LoRA [202] to fine-tune the LLM along with other trainable modules shown in Figure 4.2. The final loss is the combination of the language modeling loss and the context matching loss:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{CM}}. \tag{4.5}$$

During inference, we extract the visual tokens and all the generated `<loc>` tokens, input their hidden states through F^{vis} and F^{sum}, respectively, calculate their cosine similarities and scale them to $[0, 1]$. The start and end timestamps of the localized interval can be obtained in several ways. The simplest way is to threshold the scores with a fixed threshold and group the consecutive points whose scores are above the threshold into segments. When there are multiple segments after thresholding, a non-maximum suppression algorithm [41] can be applied to filter out overlapping predictions. We also experimented with other alternatives for extracting the segments, such as using more complex thresholding methods by capturing the critical points or appending to the LLM a learnable coordinate regression module [41]. We compare these strategies in the experiment section.

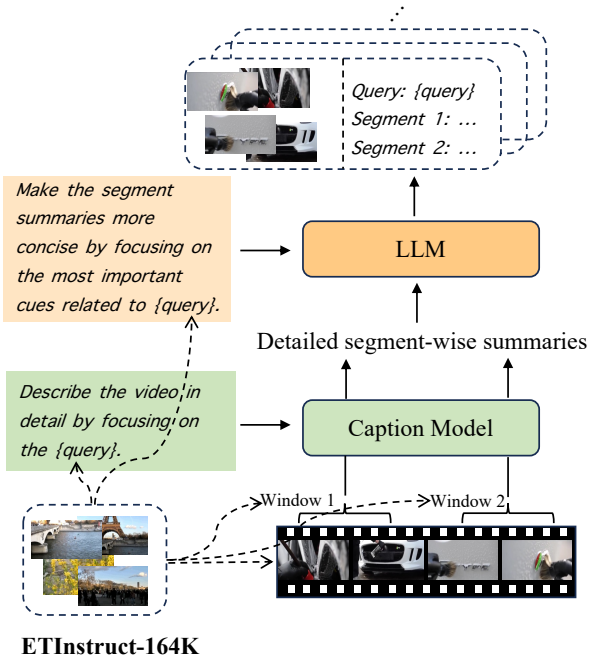

Figure 4.3: The pipeline of generating the ETSum instruction tuning dataset.

4.3.5 Instruction Fine-tuning Dataset: ETSum

Though current video LLMs have already shown powerful video temporal reasoning capability, it has been revealed in [174] that they still lack precise event-level temporal reasoning ability. To enable current video LLMs to decently handle ET video tasks purely by query-focused video summarization, we contribute an automatically created instruction fine-tuning dataset, ETSum, based on the ETInstruct dataset [174] that contains 164K videos and is collected for training video LLMs to perform ET video tasks via timestamp prediction.

The ETSum dataset inherits all the videos from ETInstruct but converts all the timestamp annotations into context-rich segment-level summaries. Specifically, we extract the ground-truth segments for each video and prompt a pre-trained VideoLLM to generate a summary for each segment conditioned on the associated query. After that, we prompt an LLM to distill long and detailed segment summaries into more concise ones while ensuring essential query-relevant contexts and coherent transitions between different ground-truth segments in the same video.

For tasks where the ground-truth segment-level captions/summaries are already available, we keep them as they are. An illustration of the data creation process is shown in Figure 4.3. The detailed statistics of the ETSum dataset are provided in the supplementary material.

4.4 Experiments

4.4.1 Dataset, Tasks, and Evaluation Metrics

We evaluate the proposed S2L on the recently proposed ETBench [174], which supports a comprehensive evaluation of video LLMs' capabilities in handling ET video tasks in four domains: referring, grounding, dense captioning, and complex temporal reasoning. Each domain involves several different fine-grained tasks. We briefly introduce each domain, its included tasks, and the associated evaluation metrics. For more detailed information, we recommend the readers refer to the original ETBench paper [174].

Referring involves referred action recognition (`[RAR]`), referred video question-answering (`[RVQ]`), and event-caption alignment (`ECA`). `[RAR]` and `[RVQ]` refer to a specific timestamp in the video and perform question answering regarding the referred place. `[ECA]` requires the model to select from a given list of time intervals the one that best matches a short query, usually an event caption. Accuracy is adopted as the evaluation metric for all three tasks. Note that as `[ECA]` requires the model to be sensitive to a list of timestamps that the proposed S2L does not support, we prompt the model to provide a context-rich summary of the clip referred by the event caption, use the context matching module to get the most confident predicted interval, and choose the one from the answer list that has the highest IoU with predicted interval as the final answer.

Grounding tasks require the model to return the temporal intervals related to a given query, usually a short event description for temporal video grounding (`[TVG]`), an ego-centric question for episodic memory (`[EPM]`), an action label for temporal action localization (`[TAL]`), a summarization instruction for extractive video summarization (`[EVS]`), and a keyword for

highlight moments for video highlight detection ([VHD]). The F1 score averaged at four levels of IoU thresholds (0.1, 0.3, 0.5, and 0.7) is used as the evaluation metric for all tasks.

Dense Captioning includes dense video captioning ([DVC]) and step localization ([SLC]). [DVC] requires the model to capture a series of major events in a video, while [SLC] requires the model to localize the steps of how-to videos. Each event or step's associated time interval needs to be returned. F1 score averaged over four IoU thresholds is adopted as the metric for the localization part, and sentence similarity is used as the metric for the captioning part. For S2L, we first parse the output into several sentences and find one or multiple predicted temporal windows for each sentence. We repeat the caption to match the number of its associated windows for evaluation.

Complex Temporal Reasoning includes temporal event matching ([TEM]) and grounded video question-answering ([GVQ]). [TEM] refers to a temporal segment in a video and requires the model to output another interval that best matches the referred one. Instead of outputting timestamps, S2L outputs a summary for the predicted segment, from which the timestamps are obtained via the context matching module. [GVQ] requires the model to answer a question and retrieve the segment that contains the answer. Recall@1 at the same IoU thresholds as those of grounding and dense captioning tasks is used as the metric for both tasks. For [GVQ], a prediction is counted as valid only if the answer is correct.

4.4.2 Implementation Details

We adopt ETChat as the backbone model, where the visual encoder is the ViT-G/14 from the pre-trained EVA-CLIP [203], and the LLM is Phi-3-Mini-3.8B [204]. Two stages of multi-modal pre-training are conducted based on the recipe from [168]. In the first stage, the frame compressor, excluding the Q-Former [46], is trained while all other components are frozen. In the second stage, the whole frame compressor is trainable, and the LLM is fine-tuned using LoRA adaptors [202]. We then use the proposed ETSum dataset to conduct instruction fine-tuning, with the context matching module with randomly initialized parameters added to the

model. During fine-tuning, the trainable modules include the attention layer, the projector in the frame compressor, the context matching module, and the newly introduced LoRA adapters to the LLM. The model is trained with FP16 mixed precision for one epoch, which takes around 10 hours on a machine with $4\times$NVIDIA A100 (80G) GPUs. All the training hyperparameters follow those applied in ETChat and are presented in the supplementary material. During the creation of ETSum, we utilized MiniCPM-V-2.6 [205] to generate segment-level summaries and GPT-4o-mini [206] to generate the ground-truth summaries.

4.4.3 ETBench Results

In this section, we will discuss the results in Table 4.1 per domain.

Referring. As the videos in the referring tasks are relatively short, the ImageLLMs only taking eight frames as input already achieve promising performance. Therefore, the video LLMs do not have an obvious advantage over the ImageLLMs on `[RAR]` and `[RVQ]` that conduct general video question and answering evaluations. However, the video LLMs significantly improve on `[ECA]` compared to the ImageLLMs, as video LLMs are capable of more precise temporal reasoning. Notably, S2L significantly outperforms all other models on `[ECA]` due to its ability to collect fine-grained contextual information in the query-related segments and the context matching module that accurately captures the shared information between the generated summary and the relevant frames.

Grounding. S2L has achieved the best results in all grounding tasks except for `[TAL]`, which requires recognition of fine-grained human actions that could be difficult to capture by relying on purely semantic cues. For `[TVG]`, S2L has a significant advantage over other models, *e.g.*, 24.7% improvement over the second best, as the `[TVG]` task requires precise understanding of event-level semantics, for which collecting more relevant context-cues could be crucial. S2L also has prominently more promising performance for `[EPM]`, `[EVS]`, and `[VHD]`, all of which require a comprehensive understanding of the queried events that the semantic-based S2L well supports. The timestamp-based models, such as TimeChat [171], VTimeLLM [175],

and LITA [172], have noticeably limited performance on such tasks, where the disadvantages of using explicit timestamps play the major role, as the embedding-based boundary prediction method, ETChat [174], has prominently better performance compared to them. However, ETChat only focuses on the boundary information while neglecting the overall event semantics, so it falls short of S2L for grounding tasks.

Dense Captioning. Though S2L has not outperformed other models as significantly on dense captioning tasks, it still holds competitive and consistent performance. We hypothesize that the reason could be that around 50% of the data in ETSum has only one segment per query, which could bias the model's ability to localize multiple segments as required by the dense captioning video. Moreover, the query-focused summaries in ETSum do not follow the concise and imperative formats of the ground-truth captions, *e.g.*, add an onion to the pan. Balancing single-segment and multi-segment data in the training set and mitigating the interference of query-focused summaries over the dense captions could be promising directions for future work.

Complex. Thanks to the context-rich output summaries, S2L can collect more evidence for handling such complex understanding tasks. As a result, S2L achieves significantly better performance than other open-source image and video LLMs and commercial MLLMS. Especially for `[GVQ]`, which requires both the correct answer and the correct localization, S2L has a 6.3% improvement over the second best. However, the absolute performance still remains unsatisfactory, leaving room for future exploration.

4.4.4 Analysis

The effect of query-focused summarization (QFS). ETChat [174] proposes to let the LLM generate special tokens (`<vid>`) whose hidden states are optimized to approximate those of the event boundary frames. In contrast to the boundary-centric approach in ETChat, we propose to guide the LLM in focusing on the semantic content of the queried event segments by requiring it to produce query-focused summaries composed of the event semantics. Thereafter, we optimize the attention-pooled hidden states of the query-focused summaries to be close to those

Table 4.1: Performance of representative MLLMs on ETBench. The best and second-best results are highlighted in green and blue, respectively.

	Referring			Grounding					Dense Captioning				Complex	
Method	RAR_{Acc}	EVC_{Acc}	RVQ_{Acc}	TVG_{F1}	EPM_{F1}	TAL_{F1}	EVS_{F1}	VHD_{F1}	DVC_{F1}	DVC_{Sim}	SLC_{F1}	SLC_{Sim}	TEM_{Rec}	GVQ_{Rec}
Random	25.0	25.0	20.0	–	–	–	–	–	–	–	–	–	–	–
Open-source ImageLLMs: 8 frames; prompts include timestamp hints.														
LLaVA-1.5 [207]	34.2	27.4	26.2	6.1	1.9	7.8	2.4	30.9	14.5	11.5	0.9	9.5	7.7	0.0
LLaVA-InternLM2 [208]	34.0	34.8	37.0	2.7	0.1	0.3	0.2	32.3	16.9	8.5	0.1	4.7	7.2	1.5
mPLUG-Owl2 [209]	37.8	26.4	34.6	1.1	0.2	3.0	4.1	36.8	0.1	8.1	0.1	7.7	6.2	0.0
XComposer [210]	33.0	19.6	40.2	4.9	1.5	9.9	2.8	28.9	5.4	5.9	2.7	9.0	10.5	0.0
Bunny-Llama3-V [211]	33.2	27.4	26.6	7.0	0.1	5.1	0.4	30.6	13.5	8.8	0.1	7.6	7.2	0.0
MiniCPM-V-2.5 [205]	37.6	28.0	37.6	2.0	0.1	4.4	13.4	18.7	6.2	11.8	1.4	9.7	0.7	0.0
Qwen-VL-Chat [212]	33.4	32.2	33.6	16.2	4.0	10.7	16.3	34.4	17.4	13.8	6.2	13.1	3.2	1.5
Open-source video LLMs: default frame counts.														
Video-ChatGPT [166]	22.6	24.2	23.0	7.0	1.3	15.1	8.4	28.8	8.8	11.3	5.7	10.2	15.9	0.0
Video-LLaVA [167]	33.6	33.0	22.6	7.0	1.9	15.0	0.3	28.9	28.0	15.0	0.9	8.3	7.5	0.1
LLaMA-VID [168]	30.4	38.4	28.8	5.5	1.2	8.0	1.4	30.0	27.1	12.6	5.2	11.1	7.0	0.9
Video-LLaMA-2 [169]	28.8	27.4	28.0	0.1	0.0	0.0	0.0	1.5	0.6	14.5	0.0	15.2	0.0	0.1
PLLaVA [213]	33.8	22.6	31.8	6.9	1.1	5.7	0.3	28.9	13.3	10.6	9.7	11.8	4.1	1.2
VTimeLLM [175]	28.4	31.0	29.2	7.6	1.9	18.2	15.9	28.9	12.4	13.1	8.7	6.4	6.8	1.9
VTG-LLM [173]	6.6	12.0	7.8	15.9	3.7	14.4	26.8	48.2	40.2	18.6	20.8	14.4	8.9	1.4
TimeChat [171]	30.8	27.6	24.6	26.2	3.9	10.1	29.1	40.5	16.6	12.5	5.6	9.2	18.0	1.5
LITA [172]	33.0	40.8	27.2	22.2	4.6	18.0	29.7	23.9	39.7	17.2	21.0	12.2	16.0	2.2
E.T.Chat† [174]	44.2	34.8	31.6	38.6	10.8	30.7	23.6	64.2	37.7	18.8	20.5	13.7	13.2	4.1
S2L (Ours)	36.4	54.2	36.2	64.3	14.8	26.9	31.1	64.9	39.4	16.0	23.3	13.9	21.9	7.8
Evaluated on 470-sample subset.														
GPT-4V [214]	33.3	40.9	46.2	27.0	1.8	18.0	28.6	55.1	16.1	19.4	21.9	13.5	23.9	0.0
GPT-4o [206]	27.8	27.3	57.7	40.4	4.5	20.0	17.6	56.9	46.9	22.3	23.1	14.9	13.6	0.0
Gemini-1.5-Flash [215]	38.9	50.0	61.5	43.9	5.4	27.0	5.4	60.8	31.6	14.9	16.5	13.3	20.8	1.0
Gemini-1.5-Pro [215]	61.1	27.3	57.7	43.1	6.2	33.8	7.9	47.0	24.0	17.5	5.8	9.8	32.1	1.0
E.T.Chat	55.6	45.5	19.2	29.7	12.5	29.0	12.6	68.7	34.9	18.2	23.1	14.7	10.5	2.1
S2L (Ours)	38.9	50.0	26.9	66.8	5.4	26.4	18.2	64.8	34.5	15.8	21.4	15.4	24.6	10.4

Table 4.2: Ablation on the effect of the query-focused summarization (QFS) task, where the metrics are reported as the average values of those from each domain's subtasks.

Method	$F1_{gnd}$	$F1_{cap}$	Sim_{cap}	Rec_{com}
ETChat	33.5	29.1	16.3	8.7
ETChat w/ QFs	30.5	26.9	18.9	7.9
S2L w/o QFS	26.7	15.0	14.2	9.4
S2L (Ours)	**40.4**	**32.0**	**19.9**	**14.9**

of the query-relevant frames instead of only boundaries. As shown in Table 4.2, optimizing the hidden states alone does not yield benefits compared to ETChat, and including the QFS task as a premise significantly boosts S2L's performance. However, as a boundary-centric approach, ETChat does not benefit from QFS. This conveys both the superiority of a semantic-based approach and the necessity of the integration of both the generative and the discriminative power of the LLM's hidden states [216] for video temporal localization.

Is the LLM necessary for a semantic-based approach? As generative models, LLMs' semantic understanding power has been mainly exploited for generative tasks, with little effort devoted to utilizing it for discriminative tasks such as video temporal localization. As contrastive vision and language models (VLMs) have been shown to excel in discriminative video tasks [41] as well, we evaluate their zero-shot performance on the grounding tasks of ETBench in Table 4.3. Indeed, such contrastive VLMS have delivered excellent performance that sometimes even surpasses those of the fine-tuned LLM-based approaches. This reinforces the conclusion that a semantic-based approach is more promising for video temporal localization. Moreover, S2L exploits the discriminative power of LLMs' hidden states via contrastive learning and achieves more consistent performance over the grounding tasks, indicating the necessity of exploiting LLM's semantic power with contrastive learning for discriminative tasks such as video temporal localization.

Is fine-tuning necessary? Though we observe that the pre-training video LLMs may also possess the query-focused summarization capability, they usually have a very low success rate of

Table 4.3: Comparison of time token generation-based models, contrastive VLMs, and S2L on grounding tasks.

Method	TVG_{F1}	EPM_{F1}	TAL_{F1}	EVS_{F1}	VHD_{F1}
Time token generation					
TimeChat [171]	26.2	3.9	10.1	29.1	40.5
VTimeLLM [175]	7.6	1.9	18.2	15.9	28.9
VTG-LLM [173]	15.9	3.7	14.4	26.8	48.2
LITA [172]	22.2	4.6	18.0	29.7	23.9
ETChat [174]	38.6	10.8	**30.7**	23.6	64.2
Semantic-based (Contrastive VLM)					
CLIP-L-14-224 [10]	35.1	10.0	19.9	30.2	62.2
EVA-G-14-224 [203]	39.7	12.7	21.7	**31.4**	61.8
SIGLIP-L-16-384 [16]	42.5	14.1	22.5	29.8	63.4
Semantic-based (LLM)					
S2L (ours)	**64.3**	**14.8**	26.9	31.1	**64.9**

Table 4.4: Comparisons of various pre-trained video LLMs and S2L on the grounding tasks. The hidden states of the pre-trained video LLMs are taken from different LLM layers, where the relative layer indices have been shown, *e.g.*, 0 stands for the first layer and 1 for the last layer.

Method	Layer Index (relative)	TVG_{F1}	EPM_{F1}	TAL_{F1}	EVS_{F1}	VHD_{F1}
MiniCPM-V-2.6	0	9.7	3.5	9.9	14.3	35.0
	0.5	10.2	4.9	9.4	8.1	27.6
	1	24.8	5.9	17.1	23.9	39.1
QWen2VL	0	9.9	3.2	8.5	16.2	36.7
	0.5	12.2	4.7	4.8	3.6	33.1
	1	30.2	4.2	14.6	26.8	46.9
InternVL2	0	12.8	3.7	12.2	16.0	33.2
	0.5	18.6	7.5	9.5	16.6	35.7
	1	26.1	7.0	13.2	23.8	37.9
S2L (Ours)	–	**64.3**	**14.8**	**26.9**	**31.1**	**64.9**

following the instruction and bring instability to their utilization. Moreover, without explicit contrastive-learning-based optimization of the hidden states, the LLMs' hidden states focus on retaining the information optimized for generation. We prompt several pre-trained video LLMs with the query-focused summarization instruction, average the hidden states of the response as the localization query hidden state like that of the `<loc>` token in S2L, and extract the segments with the threshold-based approach as elaborated in Section 4.3.4. As shown in Table 4.4, the performance of the pre-trained video LLMs, which have not been fine-tuned on the query-focused summarization and context matching tasks, lags significantly behind S2L. This indicates the necessity of fine-tuning the LLM to perform the query-focused summarization and context matching tasks.

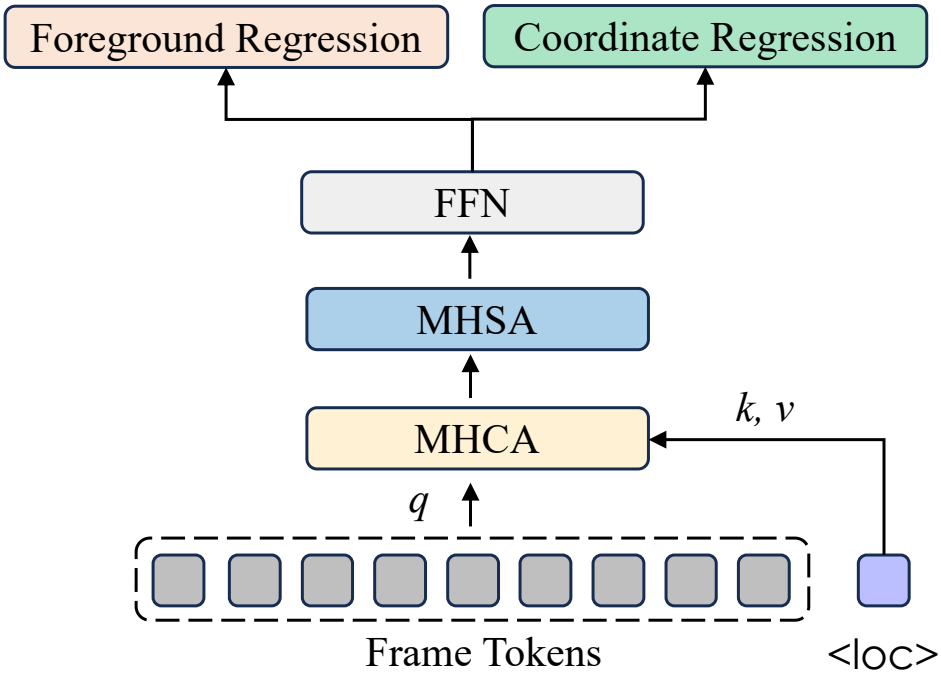

Figure 4.4: The architecture of the localization module.

Strategies of event segment mining. So far, we use the threshold-based method described in Section 4.3.4 as the default event segment mining strategy based on the cosine similarities calculated by Eq. (4.2). However, there are other valid strategies that we experimented with,

Table 4.5: Comparison of different strategies of mining the event segments given the cosine similarities.

Method	TVG_{F1}	EPM_{F1}	TAL_{F1}	EVS_{F1}	VHD_{F1}
Threshold	**64.3**	**14.8**	26.9	**31.1**	**64.9**
Critical Point	59.0	14.4	**29.9**	30.6	63.1
After adding the localization module					
Threshold	56.4	11.7	23.8	30.5	61.5
Critical Point	51.2	11.2	25.6	30.1	60.7
Critical Point & Coordinates	40.9	7.6	18.9	30.6	63.6

i.e., critical point-based strategy and coordinate head-based strategy.

For the critical point-based strategy, we first apply Gaussian smoothing to the cosine similarities and then extract all the timestamps at which local maxima are achieved. At each local maximum point, we traverse to the left and the right sides to find the nearest local minima on both sides and find the start and end timestamps of this segment anchored by the current local maximum point.

For the coordinate-based method, we append to the LLM a localization module which has a foreground regression module and a coordinate regression module [41] as shown in Figure 4.4. At the output of the localization module, each frame will have a foreground score and a start and end timestamp coordinate tuple (t^s, t^e). We only utilize the timestamp coordinates here. With the localization module, we still need to decide on which frames' output timestamp coordinates to use. We apply the critical point-based strategy to select such frames as an example to show the effect of the localization module.

As shown in Table 4.5, the critical point-based strategy yields worse performance over most of the grounding tasks, with some performance boosts only on the [TAL] task, which is the only multi-segment task. It will be an interesting future direction to explore a strategy that can strike a balance between single-segment and multi-segment tasks. Moreover, fine-tuning the LLM with the localization module hurts the performance with both the threshold-based and

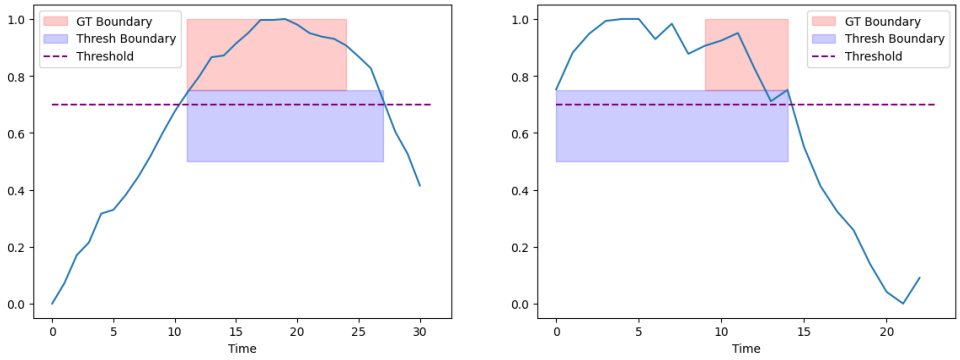

Figure 4.5: Visualization of the cosine similarities and the thresholded segments.

critical point-based strategies. Combining the critical point-based strategy with the regressed coordinates further reduces the performance. We hypothesized that the localization module involves more training parameters and thus requires more training epochs and data; the current efficient fine-tuning with only one epoch and the relatively small ETSum dataset might not be enough for the module to be well trained. Though introducing the localization module incurs computational burden, it is still interesting to explore if, given more training time and data, the combination of LLMs and an external localization module can yield promising performance that is worth the investment.

Qualitative analysis. As shown in Figure 4.5, the cosine similarities can be thresholded to get accurate predicted segments (left), but the framework still has room for improvement as the cosine similarities can be misaligned with the ground-truth event segments (right).

4.5 Conclusion

This chapter explores the potential of video LLMs for video temporal localization tasks. Different from previous works that focus on generating timestamps, we propose a query-focused summarization task to leverage the generative power of LLMs for effective discriminative learning driven by contrastive learning. As a result, the proposed S2L framework can more effectively exploit the semantic understanding capability of LLMs for video temporal localization

tasks, which has been neglected by the timestamp generation-based methods. The experimental results show that S2L significantly outperformed previous methods in most of the grounding tasks and the complex reasoning tasks, with competitive performance on dense captioning tasks. Nonetheless, S2L still struggles with multi-segment tasks. We explore several segment-mining strategies given the cosine similarities, with the finding that a more fine-grained treatment of the cosine similarities based on the critical point-based strategy can improve the multi-segment performance, though it falls short on simpler single-segment tasks. Therefore, a major future direction of this work would be to explore a more robust but still efficient segment-mining algorithm based on the current framework of integrating the generative and discriminative power of LLMs.

Chapter 5

Conclusion

This thesis has demonstrated the power and versatility of contrastive learning as a unifying principle for building foundation models across a range of vision and vision-language tasks. In Chapter 2, we introduced PixCon, a pixel-level contrastive framework to produce spatially discriminative features that can be applied to dense visual prediction tasks. In Chapter 3, we showed that training-free, zero-shot video summarization can be achieved by directly formulating classical diversity and representativeness heuristics into contrastive-score metrics in a frozen embedding space, demonstrating the power of contrastive features in training-free zero-shot applications across domains. Finally, Chapter 4 presented S2L, which integrates a generative Video LLM with a lightweight contrastive grounding module to translate free-form text summaries into precise temporal localizations, showing the potential of contrastive learning in aiding supervised learning tasks.

Looking ahead, there are several promising directions to further extend this work. First, the exploration of pixel-level learning has been limited to the convolutional neural networks, while the vision Transformers have achieved great success recently in dense visual prediction tasks. It is necessary to keep exploring the potential of contrastive learning for various downstream tasks with new architectures. Second, our zero-shot video summarizer is no more than a keyframe extraction that struggles to reflect the higher-level human intents during their summarization process. It would be interesting to further explore such training-free and zero-shot video sum-

marizers with powerful large language models. Third, though S2L leverages powerful large language models for universally handling video temporal localization tasks, it still has major performance gaps with the specialist models for the individual tasks. It would be interesting to keep pushing the performance limit of such an approach, such that it can be deployed in real-life applications.

Acknowledgements

I would like to express my deepest gratitude to Prof. Yuta Nakashima from SANKEN, the University of Osaka, Prof. Hajime Nagahara from D3 Center, the University of Osaka, and Dr. Mayu Otani from CyberAgent, Inc., for their unwavering support, invaluable guidance, and continuous encouragement throughout my doctoral studies.

I am also especially thankful to Dr. Mayu Otani for her mentorship during my internship and part-time employment at CyberAgent, Inc. I deeply appreciate her willingness to share her expertise, provide thoughtful criticism, and patiently guide me through challenging projects. Her support has played a crucial role in both my personal and professional growth.

I am profoundly grateful for the financial assistance provided by the 分野横断イノベーションを創造する情報人材育成フェローシップ. This fellowship enabled me to focus entirely on my research without the burden of financial concerns, and I truly appreciate their belief in my potential and investment in my academic development.

Finally, I would like to thank my family and my friends for their unwavering support, understanding, and companionship during the often strenuous journey of the PhD course. Their constant encouragement, patience, and care have been a tremendous source of strength and comfort, and I am deeply appreciative of all they have done to help me reach this milestone.

Reference

[1] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[3] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[4] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *CVPR*, 2015.

[5] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *AAAI*, 2018.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[12] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, 2021.

[13] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021.

[14] Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022.

[15] Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. In *CVPR*, 2022.

[16] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[17] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with

context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18082–18091, 2022.

[18] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16793–16803, 2022.

[19] Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabaleiro, Kun Qian, Xin Jing, Alexander Kathan, Bin Hu, and Bjoern W Schuller. Audio self-supervised learning: A survey. *Patterns*, Vol. 3, No. 12, 2022.

[20] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine learning for healthcare conference*, pp. 2–25. PMLR, 2022.

[21] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

[22] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021.

[23] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *NeurIPS*, 2021.

[24] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

[25] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels, 2021.

[26] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14176–14186, 2022.

[27] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 45, No. 12, pp. 15790–15801, 2023.

[28] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11175–11185, 2023.

[29] Monika Wysoczańska, Michaël Ramamonjisoa, Tomasz Trzciński, and Oriane Siméoni. Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1403–1413, 2024.

[30] Chaolei Han, Hongsong Wang, Jidong Kuang, Lei Zhang, and Jie Gui. Training-free zero-shot temporal action detection with vision-language models. *arXiv preprint arXiv:2501.13795*, 2025.

[31] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[32] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022.

[33] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, Vol. 33, pp. 18661–18673, 2020.

[34] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *Advances in neural information processing systems*, Vol. 33, pp. 12546–12558, 2020.

[35] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16291–16301, 2021.

[36] Xiangyun Zhao, Raviteja Vemulapalli, Philip Andrew Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label efficient semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10623–10633, 2021.

[37] Hritam Basak and Zhaozheng Yin. Pseudo-label guided contrastive learning for semi-supervised medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19786–19797, 2023.

[38] Wei Wu, Hao Chang, Yonghua Zheng, Zhu Li, Zhiwen Chen, and Ziheng Zhang. Contrastive learning-based robust object detection under smoky conditions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4295–4302, 2022.

[39] Jinhwan Seo, Wonho Bae, Danica J Sutherland, Junhyug Noh, and Daijin Kim. Object discovery via contrastive learning for weakly supervised object detection. In *European conference on computer vision*, pp. 312–329. Springer, 2022.

[40] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7352–7362, 2021.

[41] Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. r^2-tuning: Efficient image-to-video transfer learning for video temporal grounding, 2024.

[42] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, Vol. 34, pp. 11846–11858, 2021.

[43] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2794–2804, 2023.

[44] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 638–647, 2022.

[45] Jewook Lee, Pilhyeon Lee, Sungho Park, and Hyeran Byun. Expert-guided contrastive learning for video-text retrieval. *Neurocomputing*, Vol. 536, pp. 50–58, 2023.

[46] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

[47] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

[48] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[49] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[50] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[51] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. In *NeurIPS*, 2021.

[52] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. Dense siamese network for dense unsupervised learning. In *ECCV*, 2022.

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[54] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[55] Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. In *CVPR*, 2021.

[56] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*. PMLR, 2020.

[57] Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. In *ECCV*, 2022.

[58] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[59] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc V Gool. Revisiting contrastive methods for unsupervised learning of visual representations. In *NeurIPS*, 2021.

[60] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[61] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. In *NeurIPS*, 2020.

[62] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *NeurIPS*, 2019.

[63] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, Vol. 104, No. 2, pp. 154–171, 2013.

[64] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *IJCV*, Vol. 59, pp. 167–181, 2004.

[65] Zongshang Pang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. Revisiting pixel-level contrastive pre-training on scene images. In *WACV*, pp. 1784–1793, 2024.

[66] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, 2021.

[67] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, Vol. 88, No. 2, pp. 303–338, 2010.

[68] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[69] Zongshang Pang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. Pixcon: Pixel-level contrastive learning revisited. *Electronics*, Vol. 14, No. 8, p. 1623, 2025.

[70] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[71] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[72] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[73] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, Vol. 28, No. 2, pp. 129–137, 1982.

[74] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[75] Junqiang Huang, Xiangwen Kong, and Xiangyu Zhang. Revisiting the critical factors of augmentation-invariant representation learning. In *ECCV*, 2022.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[77] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.

[78] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[79] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV*, 2022.

[80] MMSelfSup Contributors. MMSelfSup: Openmmlab self-supervised learning toolbox and benchmark. `https://github.com/open-mmlab/mmselfsup`, 2021.

[81] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[82] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[83] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020.

[84] Mayu Otani, Yale Song, Yang Wang, et al. Video summarization overview. *Foundations and Trends® in Computer Graphics and Vision*, Vol. 13, No. 4, pp. 284–335, 2022.

[85] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.

[86] Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In *ECCV*, 2018.

[87] Tsu-Jui Fu, Shao-Heng Tai, and Hwann-Tzong Chen. Attentive and adversarial learning for video summarization. In *WACV*, 2019.

[88] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *ACCV*, 2018.

[89] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[90] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[91] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial LSTM networks. In *CVPR*, 2017.

[92] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *ECCV*, 2018.

[93] Yen-Ting Liu, Yu-Jhe Li, Fu-En Yang, Shang-Fu Chen, and Yu-Chiang Frank Wang. Learning hierarchical self-attention for video summarization. In *ICIP*, pp. 3377–3381. IEEE, 2019.

[94] Mrigank Rochan and Yang Wang. Video summarization by learning from unpaired data. In *CVPR*, 2019.

[95] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *AAAI*, 2019.

[96] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *ECCV*, 2020.

[97] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *ICLR*, 2022.

[98] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3124–3134, 2023.

[99] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 2019.

[100] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[101] Zongshang Pang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. Contrastive losses are natural criteria for unsupervised video summarization. In *WACV*, 2023.

[102] Zongshang Pang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. Unleashing the power of contrastive learning for zero-shot video summarization. *Journal of Imaging*, Vol. 10, No. 9, p. 229, 2024.

[103] Yoshimasa Takahashi, Naoko Nitta, and Noboru Babaguchi. Video summarization for large sports video archives. In *2005 IEEE International Conference on Multimedia and Expo*, pp. 1170–1173. IEEE, 2005.

[104] Dian Tjondronegoro, Yi-Ping Phoebe Chen, and Binh Pham. Highlights for more complete sports video summarization. *IEEE multimedia*, Vol. 11, No. 4, pp. 22–37, 2004.

[105] Baoxin Li, Hao Pan, and Ibrahim Sezan. A general framework for sports video summarization with its application to soccer. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, Vol. 3, pp. III–169. IEEE, 2003.

[106] Chekuri Choudary and Tiecheng Liu. Summarization of visual content in instructional videos. *IEEE Transactions on Multimedia*, Vol. 9, No. 7, pp. 1443–1455, 2007.

[107] Tiecheng Liu and John R Kender. Rule-based semantic summarization of instructional videos. In *Proceedings. International Conference on Image Processing*, Vol. 1, pp. I–I. IEEE, 2002.

[108] Tiecheng Liu and Chekuri Choudary. Content extraction and summarization of instructional videos. In *2006 International Conference on Image Processing*, pp. 149–152. IEEE, 2006.

[109] Jitao Sang and Changsheng Xu. Character-based movie summarization. In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 855–858, 2010.

[110] Chia-Ming Tsai, Li-Wei Kang, Chia-Wen Lin, and Weisi Lin. Scene-based movie summarization via role-community networks. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 11, pp. 1927–1940, 2013.

[111] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *ACM MM*, 2017.

[112] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. HSA-RNN: Hierarchical structure-adaptive RNN for video summarization. In *CVPR*, 2018.

[113] Litong Feng, Ziyin Li, Zhanghui Kuang, and Wei Zhang. Extractive video summarizer with memory augmented neural networks. In *ACM MM*, 2018.

[114] Junbo Wang, Wei Wang, Zhiyong Wang, Liang Wang, Dagan Feng, and Tieniu Tan. Stacked memory network for video summarization. In *ACM MM*, 2019.

[115] Luis Lebron Casas and Eugenia Koblents. Video summarization with LSTM and deep attention models. In *MMM*, pp. 67–79. Springer, 2019.

[116] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30, No. 6, pp. 1709–1717, 2019.

[117] Zhong Ji, Fang Jiao, Yanwei Pang, and Ling Shao. Deep attentive and semantic preserving video summarization. *Neurocomputing*, Vol. 405, pp. 200–207, 2020.

[118] Yen-Ting Liu, Yu-Jhe Li, and Yu-Chiang Frank Wang. Transforming multi-concept attention into video summarization. In *ACCV*, 2020.

[119] Jingxu Lin and Sheng-hua Zhong. Bi-directional self-attention with relative positional encoding for video summarization. In *ICTAI*, 2020.

[120] Yuan Yuan, Haopeng Li, and Qi Wang. Spatiotemporal modeling for video summarization using convolutional recurrent neural network. *IEEE Access*, 2019.

[121] Wei-Ta Chu and Yu-Hsin Liu. Spatiotemporal modeling and label distribution learning for video summarization. In *MMSP*, pp. 1–6. IEEE, 2019.

[122] Mohamed Elfeki and Ali Borji. Video summarization via actionness ranking. In *WACV*, 2019.

[123] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[124] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. SumGraph: Video summarization via recursive graph modeling. In *ECCV*, 2020.

[125] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In *ACCV*, 2016.

[126] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. In *NeurIPS*, 2021.

[127] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14867–14878, 2023.

[128] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[129] Yiyan Chen, Li Tao, Xueting Wang, and Toshihiko Yamasaki. Weakly supervised video summarization by hierarchical reinforcement learning. In *ACM MM Asia*. 2019.

[130] Zutong Li and Lei Yang. Weakly supervised deep reinforcement learning for video summarization with semantically meaningful reward. In *WACV*, 2021.

[131] Xufeng He, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Unsupervised video summarization with attentive conditional generative adversarial networks. In *ACM MM*, 2019.

[132] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[133] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[134] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[135] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[136] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[137] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, 2019.

[138] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, 2019.

[139] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020.

[140] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. Weakly-supervised temporal action localization by uncertainty modeling. In *AAAI*, 2021.

[141] Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, Vol. 32, No. 1, pp. 56–68, 2011.

[142] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *CVPR*, 2019.

[143] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, Vol. 33, No. 3, pp. 239–251, 1945.

[144] William H Beyer. *Standard Probability and Statistics: Tables and Formulae*. CRC Press, 1991.

[145] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *ECCV*, 2014.

[146] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[147] Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple pairwise ranking networks for personalized video summarization. In *ICCV*, 2021.

[148] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[149] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[150] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021.

[151] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pp. 6546–6555, 2018.

[152] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[153] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[154] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1130–1139, 2018.

[155] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In *European Conference on Computer Vision*, pp. 503–521. Springer, 2022.

[156] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, Vol. 31, pp. 5427–5441, 2022.

[157] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1049–1058, 2016.

[158] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

[159] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6847–6857, 2021.

[160] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8739–8748, 2018.

[161] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10714–10726, 2023.

[162] Leonard Bärmann and Alex Waibel. Where did i leave my keys?-episodic-memory-based question answering on egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1560–1568, 2022.

[163] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13204–13214, 2024.

[164] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*, 2019.

[165] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12934–12943, 2024.

[166] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[167] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[168] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2025.

[169] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[170] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024.

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

[172] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pp. 202–218. Springer, 2025.

[173] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv preprint arXiv:2405.13382*, 2024.

[174] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. Et bench: Towards open-ended event-level video-language understanding. *arXiv preprint arXiv:2409.18111*, 2024.

[175] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14271–14280, 2024.

[176] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*, 2024.

[177] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, Vol. 36, , 2024.

[178] Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, Vol. 36, , 2024.

[179] Boris Meinardus, Anil Batra, Anna Rohrbach, and Marcus Rohrbach. The surprising effectiveness of multimodal large language models for video moment retrieval. *arXiv preprint arXiv:2406.18113*, 2024.

[180] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Qingbin Liu, and Xi Chen. Trace: Temporal grounding video llm via causal event modeling. *arXiv preprint arXiv:2410.05643*, 2024.

[181] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, Vol. 35, pp. 24824–24837, 2022.

[182] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, Vol. 36, , 2024.

[183] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, Vol. 35, pp. 22199–22213, 2022.

[184] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1207–1216, 2019.

[185] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, et al. Mm-vid: Advancing video understanding with gpt-4v (ision). *arXiv preprint arXiv:2310.19773*, 2023.

[186] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[187] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

[188] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11888–11898, 2023.

[189] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

[190] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, Vol. 36, , 2024.

[191] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.

[192] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[193] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[194] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13504–13514, 2024.

[195] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[196] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 787–802. Springer, 2014.

[197] Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. hisum: a large-scale dataset for video highlight detection and summarization. *Advances in Neural Information Processing Systems*, Vol. 36, , 2024.

[198] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2794–2804, October 2023.

[199] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13623–13633, October 2023.

[200] Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Wenhai Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, et al. Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *arXiv preprint arXiv:2406.08394*, 2024.

[201] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589, June 2024.

[202] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[203] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19358–19369, June 2023.

[204] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, and Arash Bakhtiari et.al. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[205] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A gpt-4v level mllm on your phone, 2024.

[206] OpenAI. Hello GPT-4o, 2024.

[207] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024.

[208] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, and Pei Chu et.al. Internlm2 technical report, 2024.

[209] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13040–13051, June 2024.

[210] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition, 2023.

[211] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective, 2024.

[212] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[213] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava : Parameter-free llava extension from images to videos for video dense captioning, 2024.

[214] OpenAI. Gpt-4v(ision) System Card, 2023.

[215] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, and et.al. Damien Vincent. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

[216] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders, 2024.

List of Publications

Journal Publications (related to this thesis)

1. Pang, Zongshang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. "Unleashing the Power of Contrastive Learning for Zero-Shot Video Summarization." *Journal of Imaging 10, no. 9 (2024): 229.*

2. Pang, Zongshang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. "PixCon: Pixel-Level Contrastive Learning Revisited." *Electronics 14, no. 8 (2025): 1623.*

International Conference (related to this thesis)

1. Pang, Zongshang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. "Contrastive losses are natural criteria for unsupervised video summarization." *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2010-2019. 2023.*

2. Pang, Zongshang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. "Revisiting pixel-level contrastive pre-training on scene images." *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1784-1793. 2024.*

Domestic Conference (related to this thesis)

1. Pang, Zongshang, Mayu Otani, Yuta Nakashima. "Video Large Language Models Can Summarize to Localize." *Meeting on Image Recognition and Understanding (MIRU), poster track, 2025.*

International Conference (not related to this thesis)

1. Pang, Zongshang, Mayu Otani, and Yuta Nakashima. "Measure Twice, Cut Once: Grasping Video Structures and Event Semantics with LLMs for Video Temporal Localization." *IEEE/CVF International Conference on Computer Vision, 2025 (submitted)*