國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

結合時間特徵金字塔以釋放 Deformable DETR 於零樣本時間動作區段生成之潛力

TP²-DETR: Unlocking Deformable DETR for Zero-Shot Temporal Action Proposal Generation with Temporal Feature Pyramids

鄭雅勻

Ya-Yun Cheng

指導教授: 許永真 博士 & 鄭文皇 博士

Advisor: Jane Yung-Jen Hsu, Ph.D. & Wen-Huang Cheng, Ph.D.

中華民國 114 年 7 月

July, 2025

doi:10.6342/NTU202503736

Acknowledgements

在研究的過程中，我曾一度感到迷惘，也因此多次轉換方向。這份論文的完成，絕非我一人之力，在此誠摯向所有支持我的人表達感謝。

首先，衷心感謝指導教授許永真老師。老師總是耐心指導，給予我極大的空間去探索與嘗試，並在關鍵時刻適時引導，讓我獲益良多，也深感幸運。

同時，也感謝共同指導教授鄭文皇老師，以及口試委員吳家麟老師、楊智淵老師與陳駿丞老師，給予我專業且具體的建議，使論文內容更加完善。其中特別感謝楊老師細心審閱，並在撰寫上一步步給予我清晰的指引。

此外，感謝實驗室的夥伴們，儘管研究方向不盡相同，討論與交流總能帶來啟發；當我感到迷惘時，大家的建議給了我莫大支持，而生活上的互動也為我帶來許多歡樂。

最後，感謝家人與朋友的鼓勵，也想誠實地感謝自己。一直以來，我並不覺得自己是特別有天賦的人，很多時候只是靠著責任感與不想辜負的心情，撐過懷疑與壓力。很高興自己沒有放棄，並在過程中看到一點一滴的成長，也願未來的我持續相信自己，持續成長。

doi:10.6342/NTU202503736

摘要

在時間動作定位任務中，由於影片本身幀與幀之間變化慢，使用標準 Transformer 注意力機制時，易造成過度平滑的現象。其中一種有效的解法是引入 Deformable DETR 中的可變形注意力機制。然而，特別是在零樣本設定下，所使用的特徵多來自視覺語言模型，因缺乏直觀的時間特徵金字塔，使得現有方法難以充分發揮 Deformable DETR 在偵測短動作方面的潛力，正如其原先在圖像中對小物體偵測所展現的優勢。

為了解決此一限制，我們提出 TP²-DETR，這是一種創新的端對端架構，融合特別設計的時間特徵金字塔網路，以全面釋放 Deformable DETR 在零樣本時間動作區間生成上的潛能。我們探索了不同的 FPN 變體來更好地讓 Deformable DETR 發揮功效。而進一步為了整體系統的效率與訓練穩定性，我們設計了一個共享、輕量且具多尺度感知能力的顯著性預測頭進行早期監督，並加以多層輔助的動作區間預測頭提供深層監督訊號。

我們在 THUMOS14 與 ActivityNet1.3 資料集上進行實驗，TP²-DETR 在多數零樣本分割設定中達到最先進的表現，特別是在短動作比例較高的 THUMOS14 資料集中，在兩種常見的零樣本設定下，平均 mAP 分別提升了 5.14% 與 10.27%。上述結果顯示，我們所提出的設計能有效釋放 Deformable DETR 在零樣本時間動作區間生成任務中的潛力。

doi:10.6342/NTU202503736

關鍵字：時間動作區間生成、零樣本學習、可變形 DETR、特徵金字塔網路、短動作定位

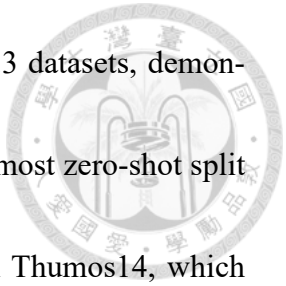
doi:10.6342/NTU202503736

Abstract

In temporal action localization, the inherent slowness of videos often leads to over-smoothing when using standard transformer attention mechanisms. A promising solution is to leverage deformable attention from Deformable DETR. However, due to the lack of an intuitive temporal feature pyramid, especially in zero-shot settings where features are extracted from vision-language models, existing methods underutilize Deformable DETR's ability to detect short actions, in the same way it benefits small object detection in images.

In this paper, we introduce TP²-DETR, a novel end-to-end framework that integrates a dedicated Temporal Feature Pyramid Network (FPN) to unlock the full potential of Deformable DETR for Zero-Shot Temporal Action Proposal Generation (ZS-TAPG). We explore different FPN variants to better leverage the capabilities of Deformable DETR. To further ensure efficiency and training stability in the end-to-end system, we design a shared, lightweight, and multi-scale-aware salient head for early supervision, complemented by auxiliary prediction heads for deep supervision.

doi:10.6342/NTU202503736



We conducted experiments on the Thumos14 and ActivityNet1.3 datasets, demonstrating that TP²-DETR achieves state-of-the-art performance across most zero-shot split settings. Notably, it yields particularly significant improvements on Thumos14, which contains a high proportion of short actions, with average mAP gains of 5.14% and 10.27% under two common zero-shot split settings. These findings demonstrate the effectiveness of our design in fully harnessing Deformable DETR for ZS-TAPG.

Keywords: Temporal Action Proposal Generation, Zero-Shot Learning, Deformable DETR, Feature Pyramid Network, Short Action Localization

doi:10.6342/NTU202503736

Contents

doi:10.6342/NTU202503736

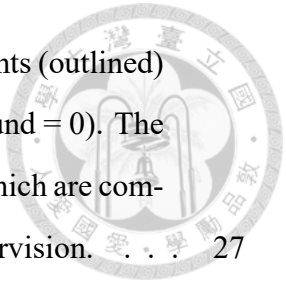
doi:10.6342/NTU202503736

doi:10.6342/NTU202503736

doi:10.6342/NTU202503736

List of Figures

doi:10.6342/NTU202503736



doi:10.6342/NTU202503736

doi:10.6342/NTU202503736

List of Tables

doi:10.6342/NTU202503736

doi:10.6342/NTU202503736

Chapter 1 Introduction

This chapter presents an overview of the thesis, beginning with the research background and motivation, followed by a summary of the proposed method, and concluding with an outline of the overall structure.

1.1 Background

In the field of video understanding, Temporal Action Localization (TAL) is a fundamental task that aims to identify where and which specific actions occur within long untrimmed videos. This capability is highly valuable for many real-world applications. Based on the definition of TAL, it can be naturally divided into two subtasks: proposal generation, which determines the temporal segments where actions happen, and proposal classification, which assigns action labels to these segments.

Over the past few years, TAL methods under the closed-set setting [21, 22, 39] have achieved remarkable progress. However, obtaining temporal annotations for long untrimmed videos is labor-intensive and expensive, which limits the scalability of fully-supervised approaches. Meanwhile, recent advances in vision-language models (VLMs) have demonstrated strong zero-shot classification capabilities, owing to their powerful semantic alignment and transferability. Consequently, Zero-Shot Temporal Action Local-

doi:10.6342/NTU202503736

ization (ZS-TAL) has gained increasing attention, with most approaches intuitively leveraging pre-trained VLMs for the classification subtask.

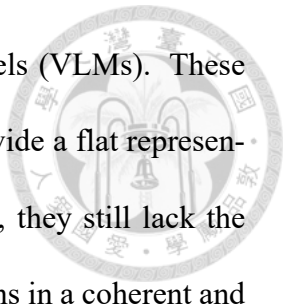
1.2 Motivation

To model temporal relationships within videos, Transformers have become a common choice due to their powerful attention mechanism. However, applying standard dense attention across long sequences of video frames often leads to an over-smoothing problem. This issue arises because video content typically changes slowly over time, and the repeated dense attention across similar frames tends to excessively average the features. As a result, the discriminability at each timestamp is weakened, making it more difficult to accurately localize action boundaries.

To address this, deformable attention has emerged as a promising solution. It introduces a sparse attention mechanism, originally proposed in Deformable DETR [40] for object detection, which focuses only on a small set of relevant positions. This sparsity helps alleviate over-smoothing and has motivated some recent studies [22, 28] to adopt Deformable DETR as the primary architecture for TAL.

While the existing approaches emphasize deformable attention itself, another key strength of Deformable DETR lies in its use of multi-scale feature pyramids, which significantly improve the detection of small objects in spatial domains. However, this advantage has not been fully leveraged in the temporal domain. This underexplored potential stems from the fact that videos do not have an explicit and natural downsampling path, unlike images, which can easily produce multi-scale feature maps through backbone networks such as ResNet [7]. This challenge is even more noticeable in zero-shot scenarios, where

doi:10.6342/NTU202503736



we rely on features extracted from pre-trained vision-language models (VLMs). These features are typically taken from the final encoder layer and thus provide a flat representation. Even if intermediate features are accessed at additional cost, they still lack the hierarchical structure needed to represent different temporal resolutions in a coherent and interpretable way, making them not well-suited to serve as a temporal pyramid.

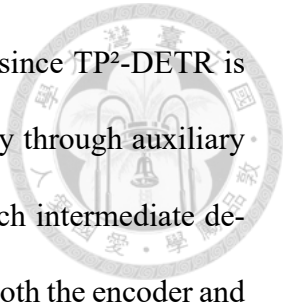
It highlights the need for a dedicated temporal feature pyramid network (FPN) that can effectively simulate a temporal scale hierarchy, allowing Deformable DETR to generalize its advantage in small-object detection to the localization of short actions in ZS-TAL scenarios.

1.3 Proposed Method

To address the aforementioned limitations and fully unlock the potential of Deformable DETR in the temporal domain, we propose **TP²-DETR** (Unlocking Deformable **DETR** for Zero-Shot **T**emporal Action **P**roposal Generation with **T**emporal Feature **P**yramids), a novel architecture designed specifically for the proposal generation subtask in ZS-TAL, which we refer to as Zero-Shot Temporal Action Proposal Generation (ZS-TAPG) for simplicity.

Building upon the insights above, TP²-DETR is designed with three key design steps as its main components. First, we design and integrate different temporal FPN variants that simulate multi-scale temporal structures, drawing inspiration from how spatial pyramids are built. Second, to enable early supervision, we adopt the use of a salient head following a prior study [4], but replace the original single-scale CNN-based version with a multi-scale-aware variant implemented as a unified MLP-based design, which is better suited

doi:10.6342/NTU202503736



to operate over the multi-scale format in Deformable DETR. Third, since TP²-DETR is built as an end-to-end system, we further improve its training stability through auxiliary supervision. Specifically, we apply additional prediction heads to each intermediate decoder layer for deep supervision, and reuse the shared salient head on both the encoder and the temporal FPN outputs to provide stronger early supervision. From a functional perspective, these three components can also be simply interpreted as: temporal FPN, deep supervision, and early supervision.

We evaluate TP²-DETR on two publicly available benchmarks: Thumos14 [9] and ActivityNet v1.3 [8]. Across almost all zero-shot split settings, our model achieves superior performance compared to previous work. Especially on Thumos14, which is dominated by short actions, TP²-DETR exhibits particularly significant improvements. These findings support our original motivation to adapt the strengths of Deformable DETR in small-object detection and extend them to short-action localization, demonstrating the effectiveness of our model design in addressing the ZS-TAPG task.

1.4 Outline of the Thesis

This thesis is organized as follows: Chapter 1 provides an overview, including the background, motivation, and a summary of the proposed method. Chapter 2 presents related work on topics relevant to our research, ranging from object detection to the temporal action localization task. Chapter 3 formally defines the problem we aim to solve. Chapter 4 details our proposed method for addressing this problem. Chapter 5 offers comprehensive experimental results to demonstrate the effectiveness of our approach. Lastly, Chapter 6 concludes the thesis by summarizing the contributions, discussing current lim-

doi:10.6342/NTU202503736

itations, and suggesting directions for future work.

doi:10.6342/NTU202503736

Chapter 2 Related Work

In this chapter, we review related work that forms the basis of our proposed method. Section 2.1 introduces object detection approaches, highlighting the transition from earlier heuristic-based designs to transformer-based models such as DETR and its variants. Section 2.2 extends the discussion to the temporal domain by covering recent developments in Temporal Action Localization (TAL), including both two-stage and one-stage frameworks. In Section 2.3, we focus on Zero-Shot Temporal Action Localization (ZS-TAL), and categorize existing methods into training-based and training-free approaches. Lastly, Section 2.4 discusses a common challenge in transformer-based TAL models, known as the over-smoothing problem, and reviews several architectural solutions that have been proposed to address this issue.

2.1 Object Detection

Object detection has long been a fundamental task in computer vision, aiming to detect and classify objects within static images. Earlier approaches such as R-CNN [5] and YOLO [30] often rely on hand-crafted anchor design and post-processing steps, such as region proposals and non-maximum suppression (NMS). Due to the limitations of these heuristic-based designs, the introduction of DETR (DEtection TRansformer) [2] brought

doi:10.6342/NTU202503736

about a significant shift in how object detection is formulated. DETR builds upon a transformer encoder-decoder architecture and reformulates object detection as a set prediction problem. It predicts a fixed-size set of objects using learnable queries and establishes one-to-one assignments with ground-truth annotations via bipartite matching. This design eliminates the need for handcrafted components and makes the model conceptually simple and end-to-end trainable.

Owing to its straightforward formulation, DETR has been widely adopted and extended by a series of follow-up studies. For instance, Deformable DETR [40] introduces deformable attention modules to reduce computational overhead and speed up convergence, while also improving detection quality, particularly for small objects. Conditional DETR [24] enhances localization capability and accelerates training by decoupling the roles of content and spatial queries. DAB-DETR [20] reformulates queries as dynamic anchor boxes that are iteratively refined to improve both convergence and localization accuracy. DN-DETR [16] further improves training efficiency by incorporating a denoising strategy that stabilizes the learning process.

2.2 Temporal Action Localization (TAL)

Temporal action localization (TAL) is a key task in video understanding, which aims to detect and classify human action segments within untrimmed videos. Existing approaches can generally be divided into two-stage methods [4, 26, 31, 39] and one-stage methods [10, 19, 22, 35]. Two-stage methods separate the process into proposal generation and proposal classification, often using dedicated modules for each. In contrast, one-stage methods handle both subtasks jointly within a unified framework.

doi:10.6342/NTU202503736

The TAL task is conceptually analogous to object detection. While object detection focuses on localizing objects in 2D spatial space, TAL localizes actions over 1D temporal sequences. Due to this structural similarity, many successful ideas from object detection have been adapted to TAL. In particular, DETR-based [4, 12–14, 22, 32, 41] and DETR-like [33, 34] architectures have been widely adopted in recent TAL research. For example, TadTR [22] builds directly upon Deformable DETR, while GAP [4] adopts Conditional DETR for generalizable zero-shot proposal generation, and several other studies propose improvements upon the vanilla DETR [14, 32, 41]. Other models, such as RTD-Net [33] and PointTAD [34], follow similar DETR-style decoding and bipartite matching, but deviate from standard DETR-based designs by removing the transformer encoder, relaxing one-to-one matching, or incorporating point-based queries and custom interaction modules.

2.3 Zero-Shot Temporal Action Localization (ZS-TAL)

Zero-Shot Temporal Action Localization (ZS-TAL) aims to train TAL models on seen action classes and expects them to generalize to unseen action classes during inference. Most ZS-TAL methods achieve this by leveraging Vision-Language Models (VLMs), which offer remarkable zero-shot generalization capabilities. Since the emergence of VLMs such as CLIP [29] in 2021, ZS-TAL has attracted increasing attention, with relevant studies beginning to appear from 2022 onward.

Early research on ZS-TAL primarily focused on training-based methods. However, as VLMs have continued to evolve in both representation power and generalization ability, recent efforts have begun to explore training-free approaches. In the following, we review

doi:10.6342/NTU202503736

relevant studies from both paradigms.

2.3.1 Training-Based Approaches

Similar to general TAL models, training-based ZSTAL methods can be categorized into two-stage and one-stage frameworks, depending on whether they implement proposal generation and proposal classification as sequential components or as a unified architecture.

- **Two-Stage Methods**

  A representative study that initiates two-stage training-based ZS-TAL is EffPrompt [10], which adopts off-the-shelf detectors such as AFSD [18] and A2Net [37] for proposal generation, and then efficiently adapts CLIP to downstream video understanding tasks through prompting strategies for proposal classification. Building on this idea, MMPrompt [11] enhances proposal generation by incorporating the optical flow modality as motion cues, and improves classification through detailed textual descriptions and visual-conditioned prompts.

  A more recent study, GAP [4], focuses specifically on improving the proposal generation stage. It addresses the issue of incomplete proposals by training with proposal-level objectives—rather than frame-level ones—through leveraging Conditional DETR as the primary architecture, and complements the dynamic modeling with static representations for further refinement. To the best of our knowledge, GAP achieves state-of-the-art performance in the ZS-TAL literature and is the most similar to our approach in terms of DETR-based architectures and proposal generation, serving as a key methodological reference in our work.

doi:10.6342/NTU202503736

- **One-Stage Methods**

  A pioneering one-stage training-based ZS-TAL method is STALE [26], which jointly performs proposal generation and classification through parallel branches. It introduces a class-agnostic representation masking mechanism and employs a consistency loss to align the two streams. UnLoc [36] adopts a unified architecture with two parallel prediction heads for proposal generation and classification. It further enhances the network with a feature pyramid and an early video-text fusion module to inject language priors during proposal decoding. ZEETAD [28] improves the one-stage framework in several aspects, such as replacing text prompt tuning (TPT), as used in STALE and EffPrompt, with an adapter-based module, and incorporating Deformable DETR to better handle ambiguous temporal boundaries during proposal generation.

2.3.2 Training-Free Approaches

With the increasing expressiveness of VLMs and the goal of avoiding bias introduced by the training process, training-free approaches have started to gain attention in recent ZS-TAL research. T3AL [17] aligns visual and textual embeddings using CLIP or CoCa [38] to obtain video-level pseudo labels, applies test-time adaptation via self-supervised projectors, and uses a captioning model (e.g., CoCa) to perform text-guided region suppression as post-processing. ZEAL [1] prompts a Large Language Model (LLM), such as GPT-4 [27], to expand action class names into detailed descriptions, which serve as queries for a Large Vision-Language Model (LVLM). The LVLM generates frame-level confidence scores, while CLIP is used to reduce the search space. FreeZAD [6] replaces general similarity scores from CoCa with the LogOIC score to enable more stable boundary eval-

doi:10.6342/NTU202503736

uations, and integrates frequency-based signals to calibrate actionness for more reliable ranking in the final localization outputs.

Overall, although training-free approaches currently underperform compared to training-based methods, they highlight the promising potential of VLMs in enabling ZS-TAL without requiring any additional training.

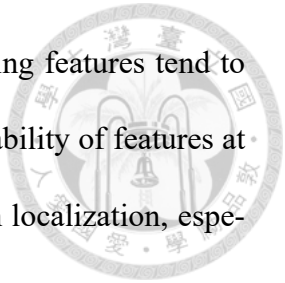
Methodological Positioning

Our proposed method belongs to the category of two-stage training-based approaches and shares a similar problem formulation with GAP [4]. Specifically, we focus on the proposal generation stage of ZS-TAL, which we refer to as *Zero-Shot Temporal Action Proposal Generation (ZS-TAPG)* for simplicity, and similarly build upon a DETR-based architecture.

2.4 Over-smoothing in Transformer-based Architectures for TAL

Transformer-based architectures have become a popular choice in TAL tasks due to their effectiveness in modeling long-range temporal dependencies. These structures are also employed in DETR-based or DETR-like models, as previously mentioned. However, applying the standard transformer attention mechanism to video data introduces a significant challenge: over-smoothing. This issue arises because standard attention uses dense aggregation, incorporating information uniformly across all time steps. At the same time, video features typically exhibit high temporal redundancy due to the slow-changing

doi:10.6342/NTU202503736



nature of visual content. When these two factors interact, the resulting features tend to become overly homogenized. Over-smoothing reduces the discriminability of features at each time step, blurs temporal boundaries, and hinders precise action localization, especially for short or rapidly changing actions.

To address this problem, recent DETR-based methods in TAL have proposed architectural modifications aimed at revising or replacing standard attention mechanisms. TranZAD [25] and RTD-Net [33] replace the transformer encoder with a multi-layer perceptron (MLP) structure in order to reduce feature mixing and preserve temporal distinctiveness. TadTR [22] and ZEETAD [28] utilize Deformable DETR as their primary architecture to specifically leverage the deformable attention mechanism, which is a form of sparse attention that selectively attends to informative regions instead of treating all positions equally, as in standard dense attention. TadTR is applied to fully-supervised TAL, while ZEETAD targets the zero-shot setting. ReAct [32] extends a similar concept by introducing a relational attention mechanism that allows the model to selectively attend to relevant action queries and preserve their discriminability.

doi:10.6342/NTU202503736

Chapter 3 Problem Statement

In the field of video understanding, Temporal Action Localization (TAL) is an important task to find meaningful segments from untrimmed videos. Because obtaining precise annotations is both time-consuming and expensive, we often only have a limited amount of labeled data available. With this constraint and the emergence of powerful vision-language models (VLMs), Zero-Shot Temporal Action Localization (ZS-TAL) has received increasing attention, which tries to detect and classify action instances in untrimmed videos without requiring any annotated examples from the target action categories. This task can be naturally decomposed into two sub-tasks: (1) **Temporal Action Proposal Generation (TAPG)**, which focuses on accurately localizing candidate segments that may contain actions of interest, and (2) **Zero-Shot Proposal Classification**, which aims to assign correct semantic labels (i.e., target action category names) to those segments.

Recent advances in VLMs, such as CLIP [29], have demonstrated remarkable capabilities in zero-shot classification by aligning visual embeddings with target text embeddings, offering an intuitive and effective off-the-shelf solution for the second sub-task. As a result, we consider that the primary bottleneck of ZS-TAL lies in the first sub-task, which we refer to as **"Zero-Shot Temporal Action Proposal Generation (ZS-TAPG)"** for simplicity.

doi:10.6342/NTU202503736

Unlike fully-supervised settings, where proposal generation can be guided by similar seen action instances, ZS-TAPG must rely on class-agnostic cues such as visual transitions or learned temporal priors, and then generalize to generate proposals for unseen actions. Discovering potential action instances is already challenging, let alone increasing the quality (i.e., precision) of the generated proposals. To address this challenge, we investigate how to further enhance existing methods to improve the quality of generating temporal action proposals under the zero-shot setting.

In the following section, we formally define the ZS-TAPG task and the notations used throughout this work. For completeness, we also include the definition of its parent task, ZS-TAL, which provides helpful context for understanding the broader problem.

3.1 Problem Defintion

3.1.1 ZS-TAL

We begin by introducing the overall ZS-TAL task, from which ZS-TAPG is derived.

Goal:

Given an untrimmed video V labeled with a set of action instances $Y = \{t_i, c_i\}_{i=1}^{i=N_{gt}}$, the model aims to predict a set of action instances $\hat{Y} = \{\hat{t}_i, \hat{c}_i\}_{i=1}^{i=N_q}$ that maximizes the evaluation metric $E(Y, \hat{Y})$.

Input:

An untrimmed video V

Output:

doi:10.6342/NTU202503736

A set of action instances $\hat{Y}$

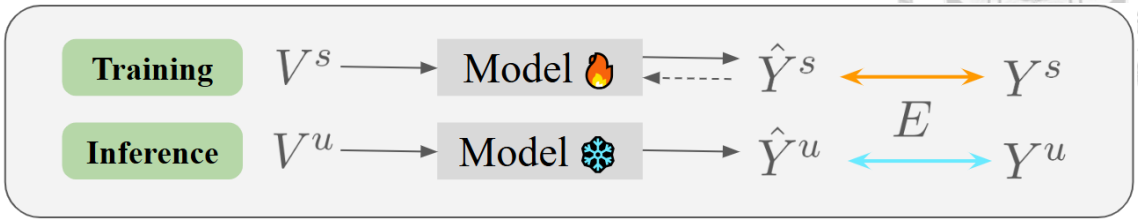


Figure 3.1: **Overview of Zero-Shot Temporal Action Localization (ZS-TAL).** During training, the model is supervised using seen videos V^s and their action instance annotations Y^s. During inference, it predicts action instances $\hat{Y}^u$ on unseen videos V^u, which are then evaluated against Y^u using the metric E.

3.1.2 ZS-TAPG

In this work, we focus exclusively on the first sub-task—**Zero-Shot Temporal Action Proposal Generation (ZS-TAPG)**. Since classification is not the target of our study, we do not rely on action category labels c_i during training or inference. Instead, the model estimates an actionness score a_i, which reflects the confidence or foreground probability of a segment containing any action (class-agnostic). These scores are then used to select and rank the final output proposals.

Goal:

Given an untrimmed video V that is labeled with a set of action proposals $P = \{t_i, a_i = 1\}_{i=1}^{i=N_{gt}}$, the model aims to predict a set of action proposals $\hat{P} = \{\hat{t}_i, \hat{a}_i\}_{i=1}^{i=N_q}$ that maximizes the evaluation metric $E(P, \hat{P})$.

Input:

An untrimmed video V

Output:

doi:10.6342/NTU202503736

A set of action proposals $\hat{P}$

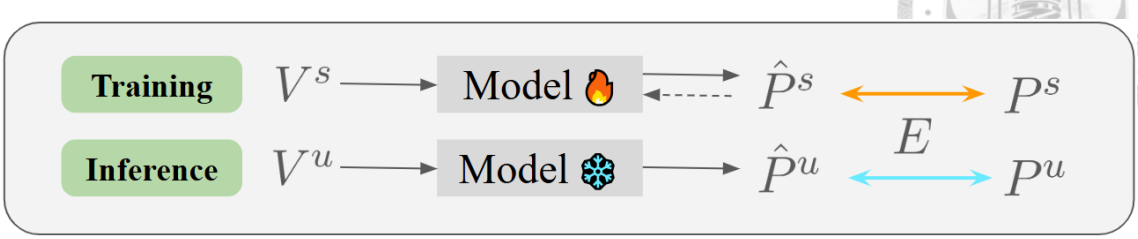


Figure 3.2: **Overview of Zero-Shot Temporal Action Proposal Generation (ZS-TAPG).** During training, the model is supervised using seen videos V^s and their action proposal annotations P^s. During inference, it predicts action proposal annotations $\hat{P}^u$ on unseen videos V^u, which are then evaluated against P^u using metric E.

3.2 Notation

- V: An untrimmed video consisting of a sequence of frames or snippets, with each snippet containing multiple consecutive frames.
- N_{gt}: The number of ground-truth action instances in the video V.
- N_q: The number of learnable queries used to generate predictions.
- $t_i = (t_i^s, t_i^e)$: The start and end timestamps of the i-th action instance/proposal.
- $c_i, \hat{c}_i$: The action category of the i-th ground-truth or predicted instance, where $c_i \in C^s$ (seen) during training and $c_i \in C^u$ (unseen) during inference.
- $a_i, \hat{a}_i$: The actionness score of the i-th ground-truth or predicted proposal, reflecting the confidence or foreground probability of a segment containing any action. Since ground-truth proposals must contain actions, $a_i = 1$ for all i.
- $Y = \{(t_i, c_i)\}$: Ground-truth annotations for action instances.
- $\hat{Y} = \{(\hat{t}_i, \hat{c}_i)\}$: Predicted action instances.

doi:10.6342/NTU202503736

- $P = \{(t_i, a_i = 1)\}$: Ground-truth annotations for action proposals.
- $\hat{P} = \{(\hat{t}_i, \hat{a}_i)\}$: Predicted action proposals.
- $E(\cdot, \cdot)$: The evaluation metric that measures task performance. In ZS-TAPG, it reflects the quality of generated proposals.

doi:10.6342/NTU202503736

Chapter 4 Methodology

In this chapter, we introduce our proposed architecture for Zero-Shot Temporal Action Proposal Generation (ZS-TAPG), which integrates a temporal feature pyramid network with Deformable DETR [40] in an end-to-end manner. Our approach aims to enhance the localization of actions, especially short actions, by leveraging the strengths of Deformable DETR in handling small objects.

For better understanding in the following sections, we first briefly review DETR-based architectures and discuss their relevance to the temporal domain.

4.1 Background

4.1.1 DETR and Deformable DETR Overview

DETR [2] reformulates object detection as a set prediction problem using transformers. Specifically, it is characterized by using a small set of learnable object queries $Q = \{q_i\}_{i=i}^{i=N_q}$, where N_q denotes the number of learnable queries, to remove the need for hand-crafted anchor design, as required by approaches like R-CNN [5]. It also adopts a bipartite matching technique via the Hungarian algorithm [15] to eliminate the need for manual post-processing such as non-maximum suppression (NMS) for dealing with near-

doi:10.6342/NTU202503736

duplicates. However, DETR suffers from slow convergence and difficulty in handling small objects. Deformable DETR was then proposed to address these limitations by introducing sparse attention (i.e., deformable attention) and incorporating multi-scale features, which significantly improve detection performance, especially for small objects.

4.1.2 From Small Objects to Short Actions

In the temporal domain, short actions can be seen as analogous to small spatial objects. However, most existing methods using Deformable DETR in TAL focus mainly on addressing the over-smoothing problem caused by dense attention, while overlooking its potential in improving the localization of short actions, which motivates us to explore how to better integrate Deformable DETR with appropriate temporal feature pyramid network (FPN) structures to enhance performance on the ZS-TAPG task.

4.2 Model Overview

The overall architecture of our proposed method is illustrated in Figure 4.1. Given an input video, instead of extracting frame-wise features, we use snippet-wise features obtained from the visual encoder of CLIP [29] for better computational efficiency. These features are then passed through a Temporal Feature Pyramid Network (Temporal FPN), which serves as the backbone to generate multi-scale temporal representations. The resulting features are encoded by a Deformable Transformer Encoder and subsequently decoded by a Deformable Transformer Decoder with a set of learnable object queries. These queries are propagated through the decoder layers to produce output embeddings, which are then passed through prediction heads (i.e., bounding box and actionness heads) to predict pro-

doi:10.6342/NTU202503736

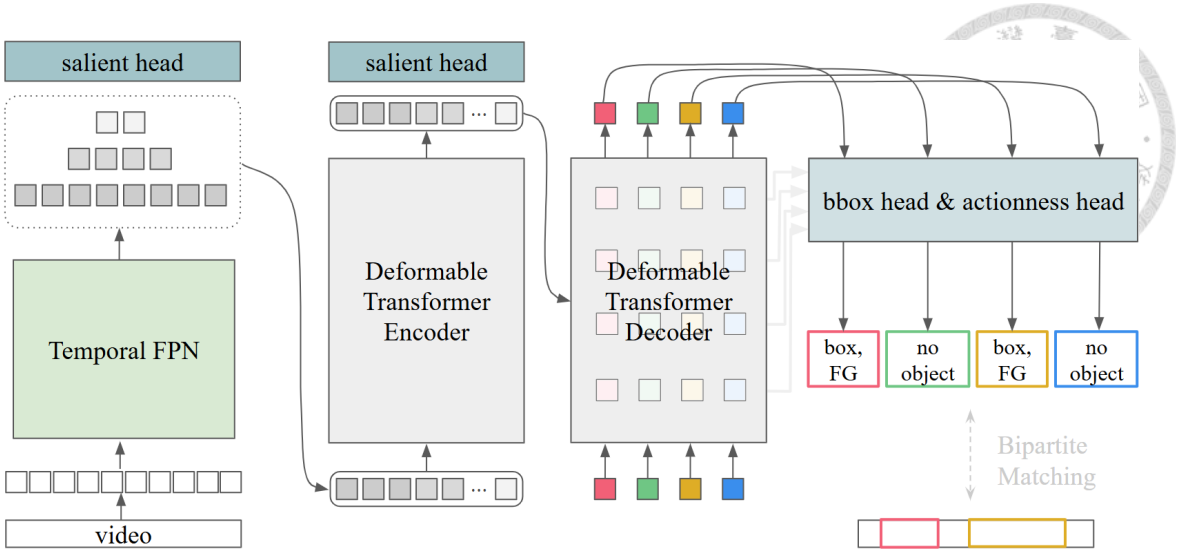


Figure 4.1: **Model Overview.** Given an input video, snippet-wise features are extracted using the CLIP visual encoder. These features are processed by a Temporal Feature Pyramid Network (Temporal FPN) to produce multi-scale temporal representations, which are then encoded and decoded by the Deformable Transformer. The model generates action proposals through prediction heads, while auxiliary salient and prediction heads are added to support stable end-to-end learning.

posal segments and their corresponding actionness scores. Throughout the pipeline, auxiliary salient heads and prediction heads are incorporated to stabilize training and encourage multi-scale temporal reasoning.

Our model introduces the following key components:

1. **Temporal Feature Pyramid Network (Temporal FPN)**: Inspired by spatial FPNs in object detection, we construct a temporal pyramid to enhance feature representation across different temporal resolutions. This structure is designed to better support Deformable DETR for short action localization. (Detailed in Section 4.3)

2. **Multi-Scale Aware Salient Head**: To complement the multi-scale design of the Deformable DETR architecture, we propose a lightweight salient head that operates on the encoder output. This head is aware of multiple temporal scales and provides early foreground-background discrimination as an auxiliary supervision signal. (Detailed in Section 4.4)

doi:10.6342/NTU202503736

3. **Auxiliary Supervision for Stable Training**: We introduce auxiliary heads on both the Temporal FPN outputs and intermediate decoder layers to provide deep supervision signals, which help stabilize the end-to-end training process. (Detailed in Section 4.5)

In the following sections, we will delve into each of the proposed components.

4.3 Temporal Feature Pyramid Network

4.3.1 Motivation

We propose to build a temporal feature pyramid to fully exploit multi-scale capability of Deformable DETR, particularly for improving the localization of short actions.

In object detection, images naturally allow for multi-scale feature maps through backbone networks such as ResNet [7], where features from the last few layers provide a spatial pyramid with increasing semantic abstraction and receptive field. However, videos do not have such an intuitive temporal downsampling path. Especially in ZS-TAL or ZS-TAPG, it is common to use vision-language models (VLMs) such as CLIP to extract frame-wise or snippet-wise features, which are typically pre-extracted and come from the final encoder layer. These features lack intermediate representations and therefore do not form a natural temporal hierarchy as ResNet does for images. Even if intermediate features are extracted from VLMs, they are not structurally aligned or semantically organized in a way that supports temporal resolution hierarchy, making them infeasible or unsuitable for direct use as a temporal pyramid. As a result, these limitations highlight the need for a dedicated temporal feature pyramid network design.

doi:10.6342/NTU202503736

4.3.2 Observation from Spatial FPN in Object Detection

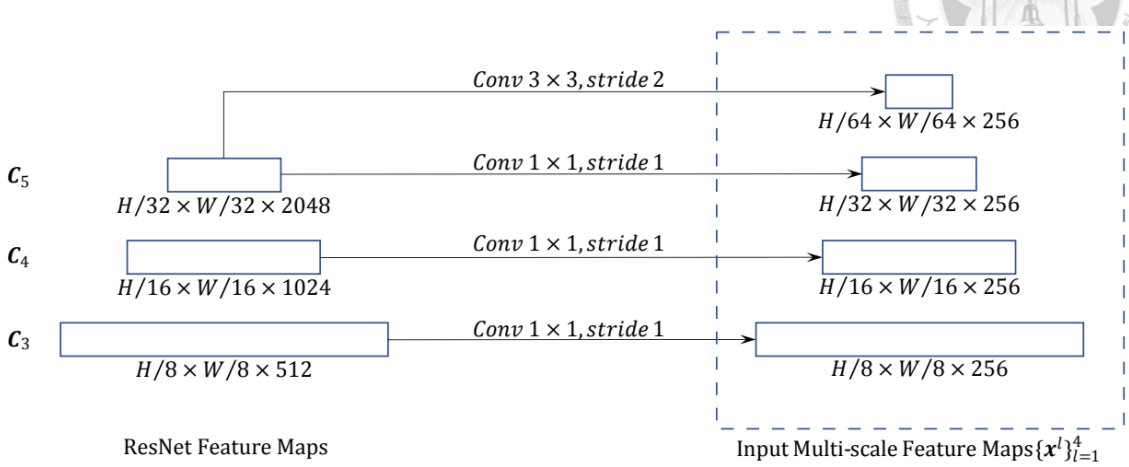


Figure 4.2: Spatial FPN used in Deformable DETR.

Before developing our temporal FPN, we first revisit the key observations from spatial feature pyramids in object detection and explore how these ideas can be extended to the temporal domain. In object detection, spatial FPNs are typically built using feature maps from the last few layers of ResNet, specifically C3 to C5, and further processed through convolution layers to align the feature dimensions (Figure 4.2). These multi-scale feature maps are expected to have the following properties:

1. **Hierarchical Resolution**: Each layer captures features at a different spatial resolution, allowing the model to handle objects of different sizes.

2. **Larger Receptive Field**: Due to stacked convolution layers and downsampling, even with the same kernel size for each layer, higher layers obtain a larger receptive field.

3. **Increasing Semantic Abstraction**: As we move from C3 to C5 (and to projected C6), not only does the resolution decrease, but the semantic level of features also increases, as deeper convolutional layers capture more high-level semantics.

doi:10.6342/NTU202503736

4. **Complementary Across Scales**: Lower layers preserve fine-grained localization, while higher layers provide coarser localization but stronger semantic context. Combining features from different scales helps improve detection performance.

4.3.3 Temporal FPN Variants

Inspired by spatial FPN, we explore three types of variants to build Temporal FPN: *Direct Downsampling*, *CNN-based Design*, and *Transformer-based Design*.

4.3.3.1 Direct Downsampling

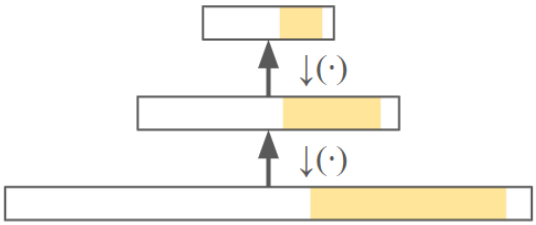


Figure 4.3: Illustration of the Direct Downsampling variant in the Temporal FPN. The yellow region indicates an action instance located within a temporal span.

The simplest way is to downsample the video feature sequence directly, level by level, thus forming a trivial pyramid of resolutions (see Figure 4.3). Although this variant does not enlarge the receptive field or enhance semantic abstraction, we still explore it due to the nature of our input features. Specifically, following prior methods [10, 26], we adopt the visual encoder F_v of CLIP to extract video representations. Given the frames of video V, we obtain feature sequences as

$$X = F_v(V) \in \mathbb{R}^{T \times C}, \tag{4.1}$$

where T denotes the number of timesteps and C is the feature dimension. Since these features are already high-level representations extracted by CLIP, we hypothesize that apply-

doi:10.6342/NTU202503736

ing additional transformations (e.g., convolutions) might potentially disrupt the original temporal structure or introduce unnecessary noise. This assumption motivates us to evaluate whether a minimal, structure-preserving design could be sufficient. For each pyramid level l we obtain

$$X^l = \downarrow (X^{l-1}) \in \mathbb{R}^{T^l \times C}, \tag{4.2}$$

where $\downarrow(\cdot)$ denotes nearest-neighbor downsampling with a fixed rate $\frac{T^{l-1}}{T^l} = 2$.

4.3.3.2 CNN-based Design

In this variant, we follow a concept similar to that used in the construction of spatial FPNs. Specifically, we build a temporal feature pyramid by applying convolution-based operations with downsampling at each level, as illustrated in Figure 4.4.

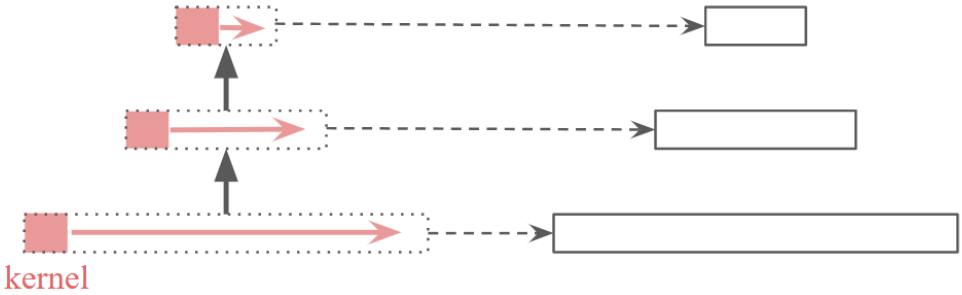


Figure 4.4: Illustration of the CNN-based Design variant in the Temporal FPN.

The basic CNN-based design mimics the standard pyramid construction in image-based backbones such as ResNet. By applying one-dimensional convolutions with a stride of 2, we downsample the temporal resolution while simultaneously expanding the receptive field. This hierarchical structure naturally provides complementary semantics across scales. Formally, starting from the initial video features X, at the pyramid level l we obtain

$$X^l = \mathrm{LN}(\mathrm{Conv}(X^{l-1})) \in \mathbb{R}^{T^l \times C}, \tag{4.3}$$

doi:10.6342/NTU202503736

where $T^l = T^{l-1}/2$, and $\mathrm{LN}(\cdot)$ denotes Layer Normalization.

Enhanced Version

The enhanced CNN variant is inspired by TriDet [31], which proposes a Transformer-like CNN architecture featuring a Scalable-Granularity Perception (SGP) block. This block incorporates both global context and local receptive fields to enhance feature discriminability. Since we specifically leverage Deformable DETR for its ability to preserve discriminability through deformable attention, we hypothesize that reinforcing this trait within the temporal FPN could lead to a synergistic effect. As a result, we adopt the SGP-based design from TriDet to construct an enhanced version of the temporal pyramid. Formally, for each pyramid level l, we obtain

$$\bar{X}^l = \mathrm{SGP}(\mathrm{LN}(X^{l-1}) + X^{l-1}), \tag{4.4}$$

$$\widetilde{X}^l = \mathrm{FFN}(\mathrm{GN}(\bar{X}^l) + \bar{X}^l), \tag{4.5}$$

$$X^l = \downarrow (\widetilde{X}^l) \in \mathbb{R}^{T^l \times C}, \tag{4.6}$$

where $\downarrow(\cdot)$ denotes max-pooling downsampling, $\mathrm{FFN}(\cdot)$ is a feed-forward network, and $\mathrm{GN}(\cdot)$ indicates Group Normalization.

4.3.3.3 Transformer-based Design

Similar to the CNN-based design, we also build a transformer-style temporal feature pyramid. In CNNs, local operations are done through convolutional kernels. In transformers, this role is naturally taken by local window self-attention, i.e., Local Multi-Head Self-Attention (Local MHSA), as shown in Figure 4.5. Specifically, we adopt a temporal

doi:10.6342/NTU202503736

FPN architecture based on the Local MHSA mechanism used in ActionFormer [39].

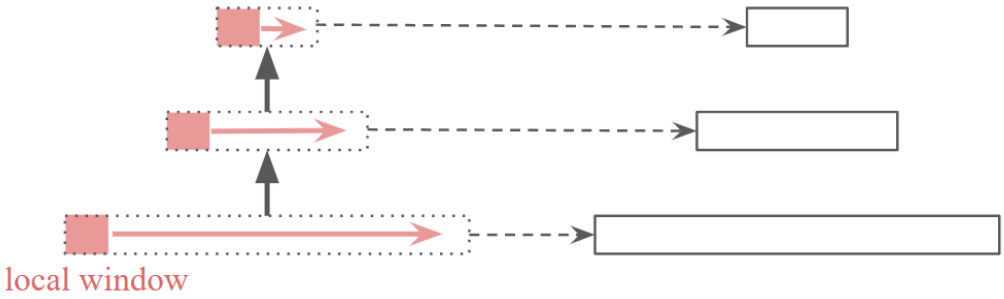


Figure 4.5: Illustration of the Transformer-based Design variant in the Temporal FPN.

Although our design is inspired by ActionFormer, we make several modifications to better suit the ZS-TAPG setting. Since our input features X are already high-level CLIP-encoded representations, we eliminate the depthwise convolutions and normalization layers that are originally applied at the beginning of each pyramid level. In addition, instead of using additional strided convolutions, we simply apply nearest-neighbor interpolation for downsampling. We make these adjustments to retain the original temporal structure and avoid introducing unnecessary transformations that might distort the feature semantics. Formally, for each pyramid level l, we obtain

$$\bar{X}^l = \alpha^l \text{LocalMHSA}(\text{LN}(X^{l-1}) + X^{l-1}), \tag{4.7}$$

$$\widetilde{X}^l = \bar{\alpha}^l \text{FFN}(\text{LN}(\bar{X}^l) + \bar{X}^l), \tag{4.8}$$

$$X^l = \downarrow (\widetilde{X}^l) \in \mathbb{R}^{T^l \times C}, \tag{4.9}$$

where $\downarrow(\cdot)$ denotes nearest-neighbor downsampling with a fixed ratio of $T^{l-1}/T^l = 2$.

doi:10.6342/NTU202503736

4.4 Multi-Scale Aware Salient Head

4.4.1 Motivation

The use of a salient head in ZS-TAPG was proposed in GAP [4], which applies it on top of the transformer encoder output to predict the foreground probability (i.e., action-ness) for each timestamp in a video. The objective is to provide an early supervision signal for the final proposal generation task. If the model can effectively distinguish foreground from background, it is more likely to localize the action segments, which correspond to the start and end timestamps of proposals. The overall process is illustrated in Figure 4.6.

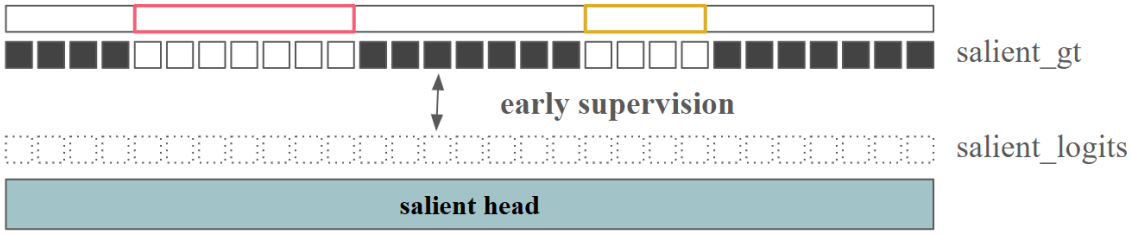


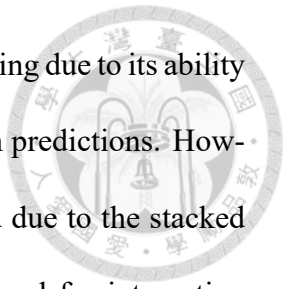
Figure 4.6: **Illustration of the salient head.** Ground-truth action segments (outlined) produce a binary mask (salient_gt, foreground = 1, background = 0). The salient head outputs per-timestamp scores (salient_logits), which are compared to salient_gt to compute the salient loss for early supervision.

4.4.2 Salient Head Types

4.4.2.1 CNN-based

In the original GAP design, the salient head is composed of 1D convolutional layers. Since GAP is built upon Conditional DETR instead of Deformable DETR, it only supports single-scale features as input. To make the salient head multi-scale aware, we extend this design by applying one salient head per scale. Each output is upsampled to a common temporal resolution, followed by a fusion step.

doi:10.6342/NTU202503736



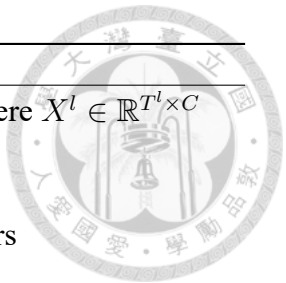
This CNN-based design benefits from improved temporal smoothing due to its ability to incorporate neighboring context, which helps reduce oscillations in predictions. However, making it multi-scale aware introduces computational overhead due to the stacked per-scale heads. Moreover, the manual fusion step may not be optimal for integrating multi-scale information.

4.4.2.2 Unified MLP-based

To alleviate the limitations of the CNN-based design, we propose a unified MLP-based salient head, as presented in Algorithm 1. Specifically, we replace convolutional layers with an MLP. Although MLPs are less capable of modeling local temporal context, they better preserve sharp boundary predictions because each neuron operates independently across timestamps. This property is particularly beneficial for short action proposals that require high boundary precision.

We concatenate the multi-scale features along the channel dimension and feed them into the MLP, allowing it to learn how to integrate different scale representations per timestamp. This design eliminates the need for manual fusion as in the CNN-based method. Additionally, the unified MLP head is more lightweight and better aligned with the output of the deformable encoder, making it more compatible in practice.

doi:10.6342/NTU202503736



Algorithm 1 Unified MLP-based Salient Head

Input: Concatenated multi-scale features $X = \text{Concat}(\{X^l\}_{l=1}^{L})$, where $X^l \in \mathbb{R}^{T^l \times C}$

Output: Salient logits $S \in \mathbb{R}^{T \times 1}$

Parameters: W_1, b_1, W_2, b_2 are weights and biases of two linear layers

$X_{\text{flat}} \leftarrow \text{Flatten}(X)$ ▷ Concat along channel dimension, $X_{\text{flat}} \in \mathbb{R}^{T \times (L \cdot C)}$

$H \leftarrow \text{ReLU}(X_{\text{flat}} \cdot W_1 + b_1)$ ▷ Cross-scale integration, $H \in \mathbb{R}^{T \times C}$

$S \leftarrow H \cdot W_2 + b_2$ ▷ Generate salient logits, $S \in \mathbb{R}^{T \times 1}$

return S

4.5 Auxiliary Heads for Stable End-to-End Learning

4.5.1 Motivation

In our current setup, the primary prediction heads (i.e., the bounding box head and the actionness head) are only used on the final output of the deformable transformer decoder. The salient head is used on the encoder output. However, since our model is designed to be an end-to-end system, this setup does not fully constrain the entire learning process. To improve training stability, we add several auxiliary heads. Specifically, we place the salient head on the output of the temporal FPN, and we also add prediction heads to each intermediate decoder layer.

4.5.2 Early Supervision on Temporal FPN

To help the model learn better features in the temporal FPN stage, we apply the salient head directly on its output. This provides early supervision signals, just like how we supervise the encoder output. Because the purpose is the same, we reuse the same shared multi-scale aware salient head introduced in the previous section.

doi:10.6342/NTU202503736

4.5.3 Deep Supervision on Decoder Layers

To make the training process more stable and efficient, we adopt a common strategy in transformer-based models by applying deep supervision. Concretely, we add independent auxiliary prediction heads to each intermediate decoder layer. This helps the model receive more direct gradient feedback at different stages of the decoder and improves overall learning quality.

4.6 Training

4.6.1 Bipartite Matching

In DETR-based models, bipartite matching via the Hungarian algorithm is a key component that enables one-to-one assignment between learnable queries and ground-truth instances. This design eliminates the need for hand-crafted post-processing.

Specifically, we assume the number of learnable queries N_q is larger than the number of ground-truth proposals N_{gt} in a video. As a result, unmatched queries are assigned to $\emptyset$. Because of the analogy between object detection and temporal action localization, and to maintain consistency with the formulation of DETR, we represent temporal action proposals—defined by their start and end timestamps $t_i = (t_i^s, t_i^e)$—using the notation b_i, following the convention of bounding boxes in object detection. The optimal assignment $\hat{\pi}$ is determined by minimizing the matching loss L_{match}, which is computed based on the temporal boundaries (i.e., $b_i = (t_i^s, t_i^e)$) and the actionness score (i.e., a_i) of each proposal predicted by the primary prediction heads (i.e., the bounding box head and the actionness

doi:10.6342/NTU202503736

head):

$$\hat{\pi} = \arg\min_{\pi} \sum_{i=1}^{N_q} L_{\text{match}}(p_i, \hat{p}_{\pi(i)}), \tag{4.10}$$

$$\begin{aligned} L_{\text{match}}(p_i, \hat{p}_{\pi(i)}) &= \mathbb{1}_{\{t_i \neq \emptyset\}}[\alpha \cdot L_{\text{bbox}} + \beta \cdot L_{\text{actionness}}] \\ &= \mathbb{1}_{\{t_i \neq \emptyset\}}[\alpha_{L1} \cdot L_{L1}(b_i, \hat{b}_{\pi(i)}) + \alpha_{\text{tIoU}} \cdot L_{\text{tIoU}}(b_i, \hat{b}_{\pi(i)}) + \beta \cdot L_{\text{focal}}(a_i, \hat{a}_{\pi(i)})]. \end{aligned} \tag{4.11}$$

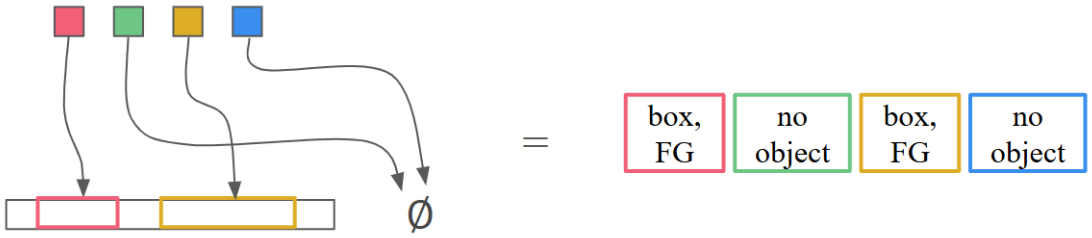


Figure 4.7: Illustration of bipartite matching.

4.6.2 Training Objectives

Given the one-to-one assignments from bipartite matching, we can define the overall training objective. The total loss consists of contributions from all prediction heads, including the bounding box heads, actionness heads, and salient heads:

$$L = \lambda_{\text{bbox}} \cdot L_{\text{bbox}} + \lambda_{\text{actionness}} \cdot L_{\text{actionness}} + \lambda_{\text{salient}} \cdot L_{\text{salient}}. \tag{4.12}$$

Here, L_{bbox} and $L_{\text{actionness}}$ are computed from all decoder layers as follows:

$$L_{\text{bbox}} = \sum_{l=1}^{N_{\text{dec}}} \sum_{i=1}^{N_q} L_{L1}(b_{l,i}, \hat{b}_{l,\pi(i)}) + L_{\text{tIoU}}(b_{l,i}, \hat{b}_{l,\pi(i)}), \tag{4.13}$$

$$L_{\text{actionness}} = \sum_{l=1}^{N_{\text{dec}}} \sum_{i=1}^{N_q} L_{\text{focal}}(a_{l,i}, \hat{a}_{l,\pi(i)}). \tag{4.14}$$

doi:10.6342/NTU202503736

The salient loss L_{salient} is calculated from both the temporal FPN and the transformer encoder outputs as

$$L_{\text{salient}} = -\sum_{t=1}^{T} L_{\text{BCE}}(m_t, s_t^{\text{fpn}}) + L_{\text{BCE}}(m_t, s_t^{\text{enc}}), \tag{4.15}$$

where λ_{bbox}, $\lambda_{\text{actionness}}$, λ_{salient} are weighting factors to balance each component.

4.7 Inference

During inference, we simply forward the video features through the network to obtain the predicted proposals, i.e., $\hat{P} = \{\hat{t}_i, \hat{a}_i\}_{i=1}^{i=N_q}$, where each proposal consists of a predicted timestamp and its corresponding actionness score, generated by the learned bounding box head and actionness head.

It is important to note that the salient head (used for early supervision) and the auxiliary prediction heads (i.e., the intermediate bounding box and actionness heads for deep supervision) are only employed during training and are omitted during inference.

To clearly illustrate the structural differences between training and inference stages, we provide two diagrams in Figure 4.8 and Figure 4.9. During training, auxiliary modules such as salient heads and intermediate prediction heads are enabled to provide both early and deep supervision, with bipartite matching used to align predictions with ground truth for loss computation. In contrast, the inference stage disables these components and directly utilizes the outputs from the final decoder layer for proposal generation.

doi:10.6342/NTU202503736

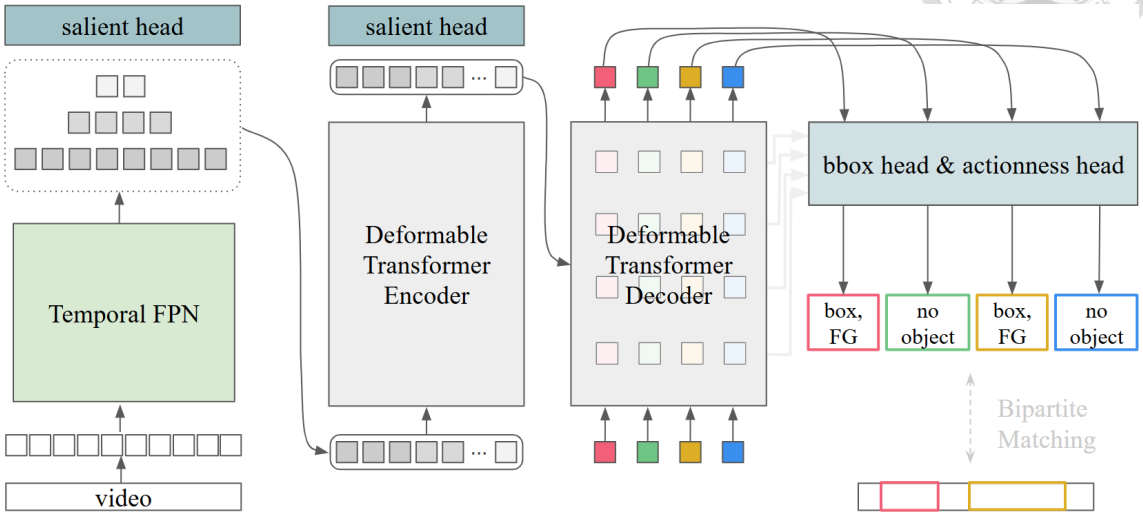


Figure 4.8: Illustration of training pipeline.

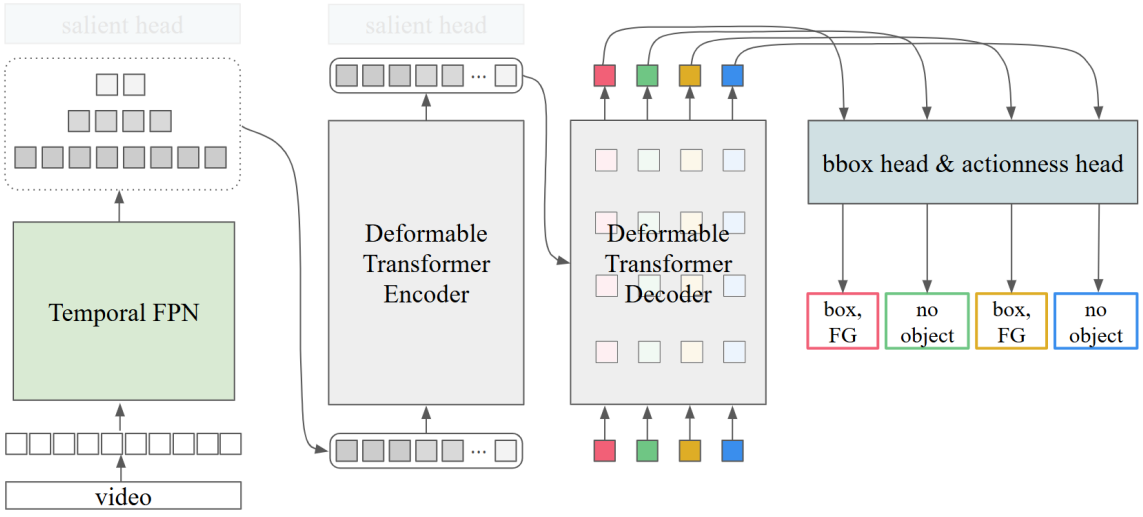


Figure 4.9: Illustration of inference pipeline.

doi:10.6342/NTU202503736

Chapter 5 Experiments

This chapter details the experimental evaluation of our proposed TP²-DETR for ZS-TAPG, a subtask of ZS-TAL. We begin by introducing the datasets, evaluation metrics, and implementation details. We then present the main results in comparison with prior work, followed by an ablation study analyzing the temporal FPN choices and design, the salient head, the contribution of each component, and qualitative visualizations highlighting improvements, especially for short actions.

5.1 Datasets

We evaluate our model on two publicly available benchmarks: Thumos14 [9] and ActivityNet1.3 [8].

5.1.1 Thumos14

The Thumos14 (Thumos 2014) dataset contains 413 untrimmed videos with temporal annotations across 20 action classes. It provides 200 validation videos (approximately 6 hours) and 213 test videos (approximately 6.5 hours). The videos are relatively short, ranging from a few seconds to a few minutes, with around 15 action instances per video on average. Following prior work, we use the validation set for training and the test set

doi:10.6342/NTU202503736

for inference in our experiments.

5.1.2 ActivityNet1.3

The ActivityNet1.3 (ActivityNet v1.3) dataset consists of 19,994 untrimmed videos collected from YouTube, covering 200 action categories. It includes 10,024 training videos (648 hours) and 4,926 validation videos (127 hours). Unlike Thumos14, ActivityNet1.3 videos are longer, typically between 5 to 10 minutes, with an average of 1.5 action instances per video. Following prior work, we use the training set for training and the validation set for inference in our experiments.

5.1.3 Zero-Shot Split Settings

To evaluate our model under zero-shot scenarios, we adopt two commonly used split settings from existing methods [4, 10, 26].

1. **75/25 Split:** 75% of the action categories are used for training, and the remaining 25% are used for testing.

2. **50/50 Split:** 50% of the action categories are used for training, and the remaining 50% are used for testing.

For both settings, we report the final performance as the average over 10 different combinations (i.e., `split_ids`), as depicted in Figure 5.1.

doi:10.6342/NTU202503736

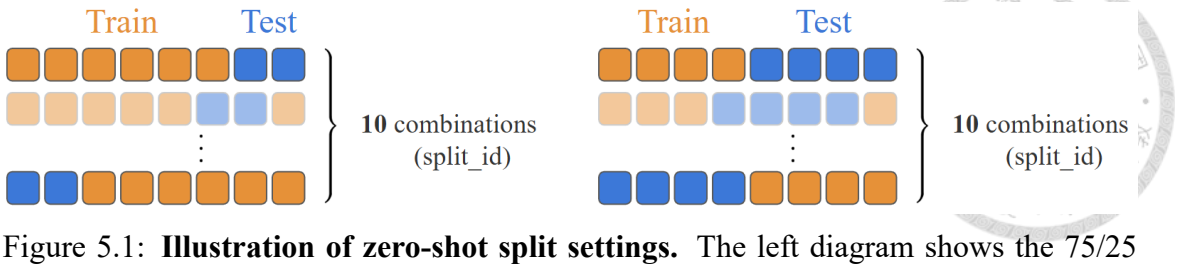


Figure 5.1: **Illustration of zero-shot split settings.** The left diagram shows the 75/25 split, and the right diagram shows the 50/50 split. Each row denotes one possible train/test split (i.e., `split_id`). Each square represents an action category.

5.2 Evaluation Metrics

In our work, we use mean Average Precision (mAP) under multiple temporal Intersection over Union (tIoU) thresholds as the main evaluation metric, a standard criterion for measuring proposal quality for temporal action localization. Typically, mAP is reported in the format **mAP@[lowest:step:highest]**, indicating the average of **Average Precision (AP)** values over multiple specified tIoU thresholds. To obtain each AP, we first sort all predicted proposals by their confidence scores (i.e., actionness score), and then match them to ground-truth segments based on the given tIoU threshold. After that, we compute the precision-recall (PR) curve and calculate the area under the curve as the AP.

In the general setting, AP is usually computed for each class independently, and the final mAP is averaged over all classes and thresholds. However, our model follows a two-stage design for zero-shot temporal action localization, where the final classification is handled by vision-language models (VLMs). As a result, we intentionally design the proposal generation stage to be **class-agnostic**, where the model only predicts whether a segment is likely to contain any action (i.e., foreground), regardless of its semantic category. This design simplifies the proposal evaluation while remaining consistent with standard mAP computation practices, as illustrated in Algorithm 2.

doi:10.6342/NTU202503736

Following the standard protocols [19], we adopt the following evaluation settings:

- **Thumos14**: mAP@[0.3:0.1:0.7]

  Averaged over 5 tIoU thresholds: {0.3, 0.4, 0.5, 0.6, 0.7}.

- **ActivityNet1.3**: mAP@[0.5:0.05:0.95]

  Averaged over 10 tIoU thresholds: {0.5, 0.55, ..., 0.95}.

Algorithm 2 Class-Agnostic mAP Computation for Proposal Generation

Input:

G: Ground-truth proposals across all videos

P: Predicted proposals with actionness scores (i.e., confidence scores)

T: Set of tIoU thresholds

Output:

mAP: Mean Average Precision across all tIoU thresholds

foreach threshold t_{IoU} in T **do**
  Sort P by actionness score in descending order
  Initialize empty set $matched\_gt$ to store matched ground truths
  foreach prediction p in P **do**
    if exists $g \in G$ in same video s.t. $g \notin matched\_gt$ and IoU$(p, g) \geq t_{IoU}$ **then**
      Mark p as True Positive (TP)
      Add g to $matched\_gt$
    end
    else
      Mark p as False Positive (FP)
    end
  end
  Compute precision and recall from TP/FP labels
  Compute $\text{AP}_{t_{IoU}}$ as area under Precision-Recall curve
end
Compute $mAP = \frac{1}{|T|} \sum_{t_{IoU} \in T} AP_{t_{IoU}}$

doi:10.6342/NTU202503736

5.3 Implementation Details

For data preparation, to ensure fair comparison with existing methods [4, 10, 26], we adopt the visual encoder from pre-trained CLIP [29] (ViT-B/16) to extract video features with per-timestep feature dimension $C = 512$. In our design, we choose to use a snippet, which aggregates 8 consecutive frames, as the timestamp unit for efficient computation.

For the temporal FPN, we set the number of pyramid layers to 4. The window size for Local MHSA is set to 9 for Thumos14 and 17 for ActivityNet1.3. Regarding the Deformable DETR [40], we use 3 encoder layers and 5 decoder layers for Thumos14, and 2 encoder layers and 3 decoder layers for ActivityNet1.3. The number of reference points for the encoder and decoder is set to 4/4 for Thumos14 and 4/6 for ActivityNet1.3. The number of queries is set to 40 and 30, respectively. Both the proposal generation heads (i.e., the bounding box head and the actionness head) are implemented using fully connected (FC) layers.

We use the AdamW [23] optimizer with a learning rate of 2×10^{-5} for the temporal FPN and 1×10^{-4} for the Deformable DETR. A multiplier of 0.1 is applied to the linear projection layers responsible for predicting reference points and sampling offsets. The weight decay is set to 1×10^{-4}. We train with a batch size of 16 for 35 epochs, using a StepLR scheduler with a decay step at epoch 30.

Key hyperparameters for bipartite matching (Eq. 4.11) are $\alpha_{L1} = 5$, $\alpha_{tIoU} = 2$, and $\beta = 2$. The loss weights (Eq. 4.12) are $\lambda_{bbox\_L1} = 5$, $\lambda_{bbox\_tIoU} = 2$, $\lambda_{actionness} = 2$, and $\lambda_{salient} = 3$ for Thumos14 and 2 for ActivityNet1.3. Our method is implemented in PyTorch and all experiments are conducted on a NVIDIA RTX 4070 GPU.

doi:10.6342/NTU202503736

5.4 Main Results

Table 5.1: **Comparison with state-of-the-art ZS-TAL methods on Thumos14 under two zero-shot splits.** AVG refers to the standard average mAP (%) metric computed across IoU thresholds [0.3:0.1:0.7], following the common evaluation protocol for Thumos14. TP²-DETR achieves state-of-the-art performance in both the 75/25 and the more challenging 50/50 splits, significantly outperforming existing methods. All methods use CLIP (ViT-B/16) with RGB frames only, without incorporating optical flow. TP²-DETR denotes our final and best-performing model, which adopts the Transformer-based design for the temporal feature pyramid. For consistency, mAP values from prior methods originally reported with only one decimal place have been padded with trailing zeros to retain two decimal places.

Split	Method	mAP@tIoU					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
75% Seen 25% Unseen	TadTR [22]‡	30.49	26.61	21.98	16.53	11.08	21.34
	EffPrompt [10]	39.70	31.60	23.00	14.90	7.50	23.30
	STALE [26]	40.50	32.30	23.50	15.30	7.60	23.80
	ZEETAD [28]§	47.30	–	29.70	–	11.50	29.70
	GAP [4]	52.30	44.20	32.80	22.40	12.40	32.90
	GAP [4]†	43.63	34.48	24.75	14.90	7.50	24.86
	GAP [4]‡	52.50	42.69	30.78	18.71	9.00	30.72
	TP²-DETR	**57.81**	**49.36**	**38.80**	**27.91**	**16.34**	**38.04**
50% Seen 50% Unseen	TadTR [22]‡	28.82	24.22	18.91	13.41	8.41	18.76
	EffPrompt [10]	37.20	29.60	21.60	14.00	7.20	21.90
	STALE [26]	38.30	30.70	21.20	13.80	7.00	22.20
	GAP [4]	44.20	36.00	27.10	15.10	8.00	26.10
	GAP [4]†	37.38	29.39	20.86	12.89	6.03	21.36
	GAP [4]‡	50.00	40.06	28.64	17.08	8.04	29.11
	TP²-DETR	**55.95**	**47.82**	**37.34**	**25.89**	**14.84**	**36.37**

† reproduced ‡ w/o class_logits § w/o extra information (i.e., optical flow) fully-supervised

5.4.1 Comparison with State-of-the-Art Methods

In Tables 5.1 and 5.2, we compare our proposed method, TP²-DETR, with existing ZS-TAL approaches on Thumos14 and ActivityNet1.3 under various zero-shot split settings. The results are reported in terms of average mAP (denoted as **AVG**). In most sce-

doi:10.6342/NTU202503736

Table 5.2: **Comparison with state-of-the-art ZS-TAL methods on ActivityNet1.3 under two zero-shot splits.** AVG refers to the standard average mAP (%) metric computed across IoU thresholds [0.5:0.05:0.95], following the common evaluation protocol for ActivityNet1.3. TP²-DETR remains competitive in the 75/25 split and shows clear improvements in the 50/50 split. All methods use CLIP (ViT-B/16) with RGB frames only, without incorporating optical flow. TP²-DETR denotes our final and best-performing model, which adopts the Transformer-based design for the temporal feature pyramid. For consistency, mAP values from prior methods originally reported with only one decimal place have been padded with trailing zeros to retain two decimal places.

Split	Method	mAP@tIoU			
		0.5	**0.75**	**0.95**	**AVG**
75% Seen 25% Unseen	TadTR [22]‡	37.59	23.34	2.97	23.06
	EffPrompt [10]	37.60	22.90	3.80	23.10
	STALE [26]	38.20	25.20	6.00	24.90
	ZEETAD [28]§	45.50	28.20	6.30	28.40
	GAP [4]	47.60	**32.50**	**8.60**	**31.80**
	GAP [4]†	44.14	29.93	6.65	29.00
	GAP [4]‡	46.91	29.08	4.65	28.81
	TP²-DETR	**48.41**	31.28	7.06	31.05
50% Seen 50% Unseen	TadTR [22]‡	35.59	21.45	2.30	21.37
	EffPrompt [10]	32.00	19.30	2.90	19.60
	STALE [26]	32.10	20.70	5.90	20.50
	GAP [4]	41.60	26.20	6.10	26.40
	GAP [4]†	40.44	27.32	5.33	26.52
	GAP [4]‡	45.67	28.53	3.97	28.18
	TP²-DETR	**47.18**	**30.25**	**6.28**	**30.08**

† reproduced ‡ w/o class_logits § w/o extra information (i.e., optical flow) fully-supervised

narios, TP²-DETR achieves state-of-the-art performance. On Thumos14, our method outperforms the previous best-performing method GAP [4] by a significant margin. Specifically, TP²-DETR improves the average mAP by 5.14% under the 75/25 split and achieves an even larger gain of 10.27% under the more challenging 50/50 split. On ActivityNet1.3, TP²-DETR remains competitive in the 75/25 setting and surpasses GAP by 3.68% in the 50/50 split.

To ensure a fair comparison, we further examine the results under settings where the semantic classification branch in GAP is removed (denoted as GAP‡), which aligns with

doi:10.6342/NTU202503736

our model design that does not rely on class-specific supervision. In this case, TP²-DETR consistently outperforms GAP‡ across all split settings, highlighting its robust generalization in the zero-shot scenario. All results are based on visual features extracted from CLIP (ViT-B/16) using RGB frames only, without incorporating additional modalities such as optical flow. Under this constraint, ZEETAD§ refers to a re-evaluated variant reproduced within the GAP framework.

Importantly, significant improvements on Thumos14 are particularly notable. Our initial motivation was to unlock the potential of Deformable DETR for short action localization by integrating a temporal FPN structure. Since a large proportion of actions in Thumos14 are short, as shown in our dataset analysis (see Figures 5.2 and 5.3), our method is especially effective on this benchmark. This validates our design assumption and explains why our method particularly excels on this dataset.

5.4.2 Comparison with Deformable DETR-based methods

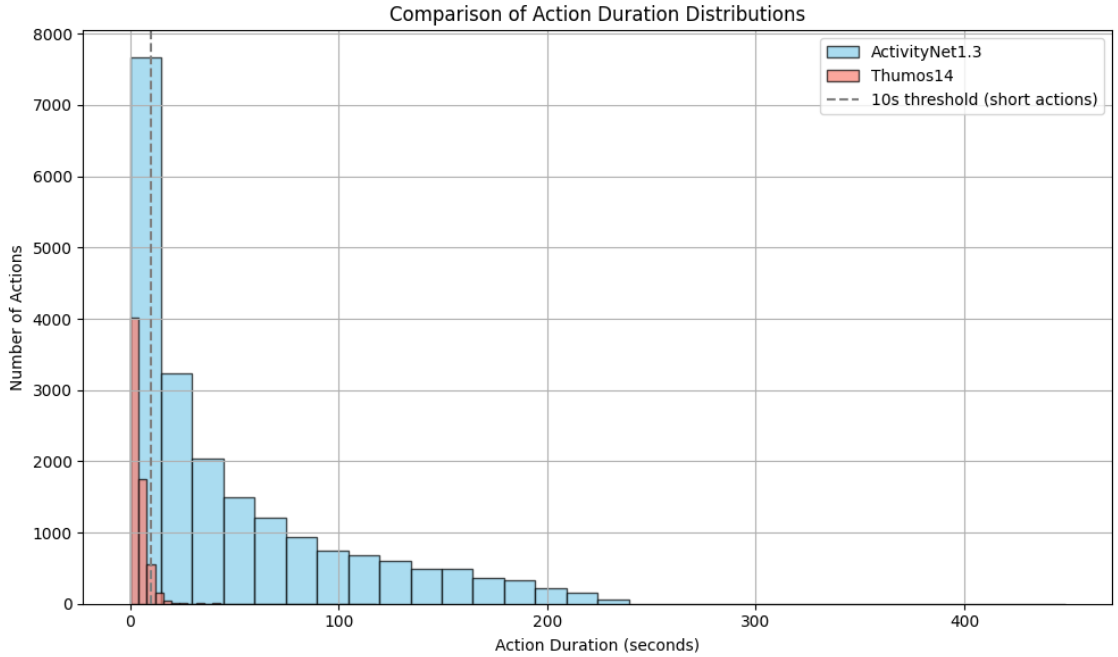
In the field of temporal action localization, two representative methods have adopted Deformable DETR as their main architecture, primarily due to the use of the deformable attention mechanism: the fully-supervised TadTR [22] and the zero-shot ZEETAD [28].

To ensure a fair zero-shot comparison, we reimplement TadTR by disabling its classification head and removing the computation and training involving the classification loss, and use actionness scores (instead of class logits) for confidence prediction, which is consistent with the setup in TP²-DETR.

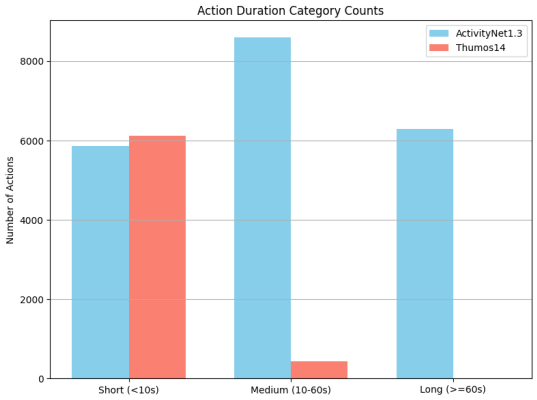
As shown in Tables 5.1 and 5.2, TP²-DETR consistently outperforms both TadTR and ZEETAD across all split settings. This highlights the effectiveness of our temporal feature

doi:10.6342/NTU202503736

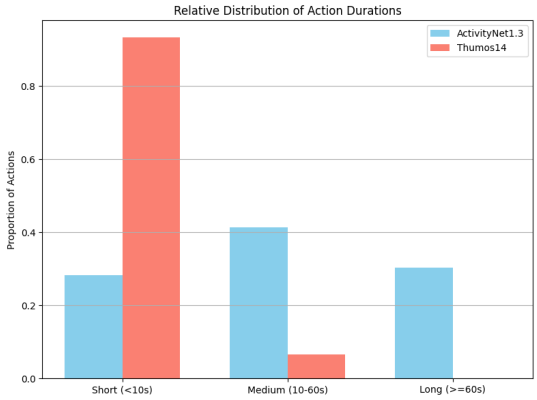
pyramid: When integrated properly, it significantly enhances the capacity of Deformable DETR in zero-shot temporal action proposal generation.



(a) Histogram of action durations (seconds).



(b) Number of actions by duration category.



(c) Relative proportion of action durations.

Figure 5.2: **Absolute view of action duration distributions on Thumos14 and ActivityNet1.3.** (a) Histogram of action durations in seconds. (b) Count of actions in short, medium, and long duration categories. (c) Relative proportions by duration category. Thumos14 contains a large number and proportion of short actions (less than 10 seconds), aligning with focus of our model on short-action localization.

doi:10.6342/NTU202503736

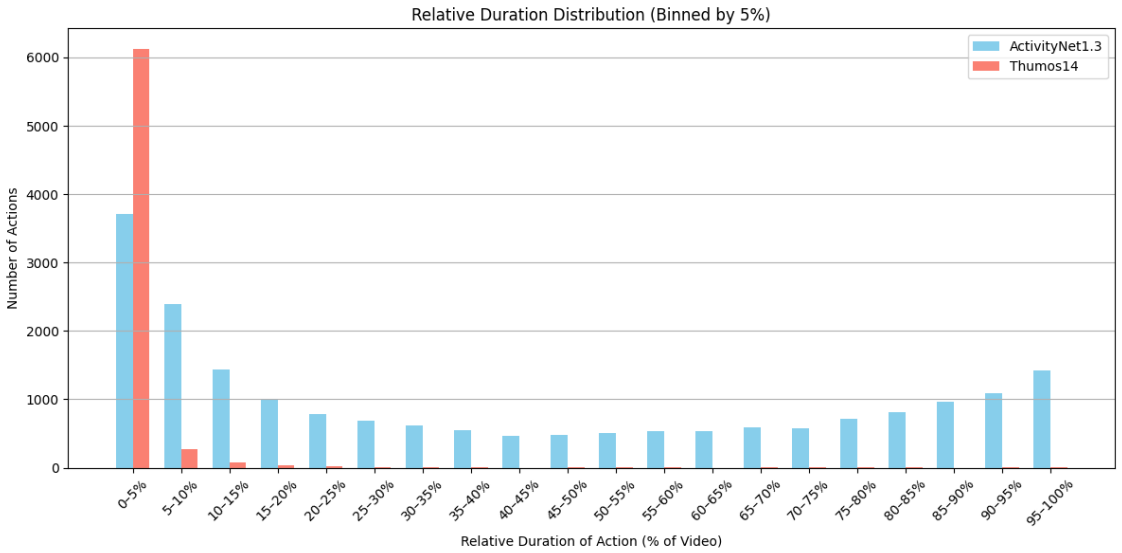


Figure 5.3: **Relative view of action duration distributions on Thumos14 and ActivityNet1.3.** The x-axis indicates the relative duration of each action instance as a percentage of the corresponding video length. Thumos14 shows a high concentration of short actions, with the majority falling under 5% of video length. This observation aligns with our motivation to enhance short-action localization.

5.5 Further Analysis and Ablation Study

To extensively evaluate our proposed TP²-DETR framework, we conduct a series of comprehensive ablation studies. Specifically, we examine the impact of different variants of the temporal FPN (Section 5.5.1), the internal design of the temporal FPN (Section 5.5.2), and the design of the salient head (Section 5.5.3). We also analyze the contribution of each individual component to overall performance (Section 5.5.4). In addition, we provide qualitative visualizations (Section 5.5.5) and analyze prediction distributions to assess the quality of model predictions (Section 5.5.6). These analyses collectively validate the effectiveness and robustness of our proposed approach.

Unless otherwise stated, all experiments and visualizations in this ablation study section are conducted on the Thumos14 dataset under the 50/50 split configuration.

doi:10.6342/NTU202503736

5.5.1 Choice of Temporal FPN

As described in Section 4.3, we explore different variants of the temporal FPN to identify the most effective structure to fully leverage the strengths of the Deformable DETR. The experimental results are summarized in Table 5.3 for Thumos14 and Table 5.4 for ActivityNet1.3.

On Thumos14, the Transformer-based variant consistently outperforms the other designs by a notable margin, followed by the Direct Downsampling variant. In contrast, on ActivityNet1.3, the enhanced CNN-based variant achieves the best performance, with the Transformer-based variant showing competitive results. Taking into account both benchmarks, we adopt the Transformer-based variant as the final design of TP²-DETR, as already presented in our main experimental results.

Table 5.3: **Analysis of different variants of the temporal FPN on Thumos14.** We compare *Direct Downsampling*, *CNN-based (basic/enhanced)*, and *Transformer-based* designs. All variants use the same overall architecture, differing only in the temporal FPN design.

Split	Type	mAP@tIoU					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
75% Seen 25% Unseen	Direct Downsampling	55.69	47.05	35.34	24.21	13.37	35.13
	CNN-based (basic)	53.46	44.65	34.88	22.86	12.09	33.59
	CNN-based (enhanced)	51.95	43.05	31.48	21.21	12.14	31.96
	Transformer-based	**57.81**	**49.36**	**38.80**	**27.91**	**16.34**	**38.04**
50% Seen 50% Unseen	Direct Downsampling	53.86	44.79	33.98	22.38	11.87	33.37
	CNN-based (basic)	51.17	42.22	31.37	19.66	9.73	30.83
	CNN-based (enhanced)	51.71	42.40	31.09	20.31	11.07	31.32
	Transformer-based	**55.95**	**47.82**	**37.34**	**25.89**	**14.84**	**36.37**

doi:10.6342/NTU202503736

Table 5.4: **Analysis of different variants of the temporal FPN on ActivityNet1.3.** We compare *Direct Downsampling*, *CNN-based (basic/enhanced)*, and *Transformer-based* designs. All variants use the same overall architecture, differing only in the temporal FPN design.

Split	Type	mAP@IoU			
		0.5	**0.75**	**0.95**	**AVG**
75% Seen 25% Unseen	Direct Downsampling	47.30	30.22	6.59	30.09
	CNN-based (basic)	47.81	30.38	6.58	30.37
	CNN-based (enhanced)	**48.62**	**31.32**	**7.15**	**31.17**
	Transformer-based	48.41	31.28	7.06	31.05
50% Seen 50% Unseen	Direct Downsampling	45.78	28.67	5.58	28.67
	CNN-based (basic)	46.29	29.22	5.64	29.12
	CNN-based (enhanced)	**48.18**	**30.68**	**6.37**	**30.46**
	Transformer-based	47.18	30.25	6.28	30.08

5.5.2 Design of Temporal FPN

Building upon the Transformer-based variant identified in the previous section, we further refine its internal design by drawing inspiration from ActionFormer [39]. However, to better accommodate the ZS-TAPG setting in TP²-DETR, we make specific adjustments to the original ActionFormer structure. To validate the effectiveness of our modified design, we present experimental results in Table 5.5 for Thumos14 and Table 5.6 for ActivityNet1.3.

On Thumos14, our Transformer-based design slightly outperforms the original ActionFormer configuration, demonstrating the benefit of these refinements under short-action-dominant scenarios. On ActivityNet1.3, however, the performance gains are less consistent across different zero-shot splits. We hypothesize that the impact of temporal FPN design is more noticeable for short actions, which are common in Thumos14 but less so in ActivityNet1.3. This observation suggests that carefully crafted temporal FPN structures are more influential in datasets dominated by short actions.

doi:10.6342/NTU202503736

Table 5.5: **Analysis of the internal design of the temporal FPN on Thumos14.** We compare the original ActionFormer design with our Transformer-based variant adapted for ZS-TAPG.

Split	Design	mAP@tIoU					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
75% Seen 25% Unseen	ActionFormer	**58.09**	**49.92**	38.79	27.14	15.57	37.90
	Transformer-based	57.81	49.36	**38.80**	**27.91**	**16.34**	**38.04**
50% Seen 50% Unseen	ActionFormer	55.92	47.00	36.22	24.11	13.20	35.29
	Transformer-based	**55.95**	**47.82**	**37.34**	**25.89**	**14.84**	**36.37**

Table 5.6: **Analysis of the internal design of the temporal FPN on ActivityNet1.3.** We compare the original ActionFormer design with our Transformer-based variant adapted for ZS-TAPG.

Split	Design	mAP@tIoU			
		0.5	**0.75**	**0.95**	**AVG**
75% Seen 25% Unseen	ActionFormer	**48.57**	**31.33**	6.67	**31.13**
	Transformer-based	48.41	31.28	**7.06**	31.05
50% Seen 50% Unseen	ActionFormer	**47.40**	30.03	6.03	29.94
	Transformer-based	47.18	**30.25**	**6.28**	**30.08**

5.5.3 Design of Salient Head

As described in Section 4.4, we adopt a unified MLP-based salient head applied to the encoder output, instead of using the multi-scale-aware CNN-based extension from GAP [4], which was originally designed for single-scale features. To assess the effectiveness of our design, we conduct experiments as shown in Table 5.7.

For a comprehensive comparison, we evaluate different fusion strategies applied to the per-scale CNN-based salient head, including max pooling (i.e., selecting the most dominant signal across levels), mean pooling (i.e., averaging outputs from all levels), and adaptive fusion (i.e., applying learnable weights to each level). Among these, the max fusion strategy achieves the best result with an average mAP of 35.91. However, regardless of the fusion method, all CNN-based designs fall short of the performance achieved by our

doi:10.6342/NTU202503736

unified MLP-based variant, which is adopted in TP²-DETR.

Furthermore, as described in Section 4.5, we extend the application of the unified salient head beyond the encoder, applying it jointly to both the encoder and the temporal FPN outputs. This is intended to enhance early supervision and better guide the proposal learning. The corresponding results are summarized in Table 5.7. Compared to applying the salient head only on the encoder output, the joint configuration yields consistently better performance, further validating the benefits of our proposed early supervision strategy.

Table 5.7: **Analysis of the design of the salient head.** We compare various feature fusion types for the per-scale CNN-based design and compare with our unified MLP-based variant.

Design	Fusion Type	mAP@tIoU					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
Per-scale CNN-based	max	55.74	47.49	36.86	25.05	14.43	35.91
Per-scale CNN-based	mean	**56.47**	47.68	36.64	24.51	13.26	35.71
Per-scale CNN-based	adaptive	55.89	47.24	36.22	24.47	13.87	35.57
Unified MLP-based	-	55.95	**47.82**	**37.34**	**25.89**	**14.84**	**36.37**

Table 5.8: **Analysis of the placement of the unified MLP-based salient head.** We compare applying it solely on the encoder (enc) versus jointly on both the encoder and temporal FPN outputs (fpn + enc).

Design	Upon	mAP@tIoU					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
Unified MLP-based	enc	55.02	47.06	36.59	25.17	14.72	35.71
	fpn + enc	**55.95**	**47.82**	**37.34**	**25.89**	**14.84**	**36.37**

5.5.4 Effectiveness of Components

To further assess the effectiveness of each proposed component in our framework, we conduct a set of ablation experiments summarized in Table 5.9. Our model introduces three core components: (1) the temporal FPN, (2) a multi-scale-aware salient head, and (3) auxiliary supervision for training stability. The auxiliary supervision includes both

doi:10.6342/NTU202503736

deep supervision via additional decoder-layer prediction heads and early supervision via a shared salient head applied to earlier stages.

Notably, since the salient head is functionally involved across two proposed components, we avoid defining the ablation as a simple additive combination. Instead, we restructure the experiment based on distinct functional objectives to better isolate each contribution. Specifically, we divide the experiments into four configurations as follows:

Baseline: A Deformable DETR model using single-scale features (no temporal FPN), trained with the original single-scale CNN-based salient head from GAP.

+ Temporal FPN (Tx): The baseline model with a Transformer-based temporal FPN, trained with a multi-scale-aware CNN-based salient head adapted from GAP.

+ Deep Supervision: Builds on the above by introducing auxiliary prediction heads (i.e., bounding box and actionness heads) for each decoder layer.

+ Early Supervision: Builds on all prior components and replaces the salient head with a unified MLP-based variant applied to both the encoder and the temporal FPN outputs, providing stronger early supervision.

From the results in Table 5.9, we observe that the introduction of the temporal FPN alone brings a noticeable improvement over the baseline across all tIoU thresholds, increasing the average mAP from 30.58 to 33.18. Incorporating deep supervision further improves the performance, raising the average mAP to 35.63. With the addition of early supervision through the unified MLP-based salient head, the model achieves a further gain of 0.74, reaching an average mAP of 36.37.

doi:10.6342/NTU202503736

The consistent performance improvements observed across all configurations validate the effectiveness and necessity of our proposed components in enhancing the quality of class-agnostic temporal action proposals, ultimately enabling TP²-DETR to achieve superior performance over existing state-of-the-art methods.

Table 5.9: **Ablation study on the effectiveness of each proposed components.** We incrementally add *Temporal FPN (Tx)*, *Deep Supervision*, and *Early Supervision* on top of the baseline and report results under different IoU thresholds. Tx indicates the Transformer-based design variant of our temporal feature pyramid.

Method	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**AVG**
Baseline	51.37	42.55	31.12	19.09	8.78	30.58
+ Temporal FPN (Tx)	53.90	44.69	33.82	22.45	11.03	33.18
+ Deep Supervision	55.96	47.18	36.37	24.84	13.80	35.63
+ Early Supervision	55.95	47.82	37.34	25.89	14.84	36.37

5.5.5 Qualitative Results

In addition to the quantitative results presented in previous sections, we also provide qualitative visualizations to further demonstrate the effectiveness of TP²-DETR, as shown in Figure 5.4. Specifically, we select the top-k class-agnostic proposals in two ways for visualization: (1) by highest actionness scores, following the GAP approach; and (2) by lowest bipartite matching cost, consistent with the matching process in the training stage. Here, k is set to the number of ground-truth action instances (N_{gt}) in each video.

The visualizations show that TP²-DETR consistently produces more accurate and comprehensive coverage of ground-truth segments compared to both GAP and the baseline described in Section 5.5.4. For clearer comparison, we additionally report the average IoU (AvgIoU) of the top-k proposals to provide an intuitive understanding of the localization quality.

doi:10.6342/NTU202503736

5.5.6 Prediction Quality

To further assess the quality of the predicted proposals, we visualize the distribution of confidence scores versus the best tIoU with ground-truth segments. Specifically, we compare the prediction distributions of TP²-DETR and the baseline model, as shown in Figure 5.5. In these scatter plots, each point represents a predicted proposal, with its horizontal position indicating the confidence score (i.e., actionness score) and the vertical position representing its best temporal IoU with the ground truth. As high-quality proposals typically appear in the top-right region (indicating both high confidence and accurate localization), TP²-DETR demonstrates a significantly denser distribution in that area compared to the baseline. This highlights the effectiveness of our temporal FPN design in improving proposal quality, especially for short actions, as evident on Thumos14.

doi:10.6342/NTU202503736

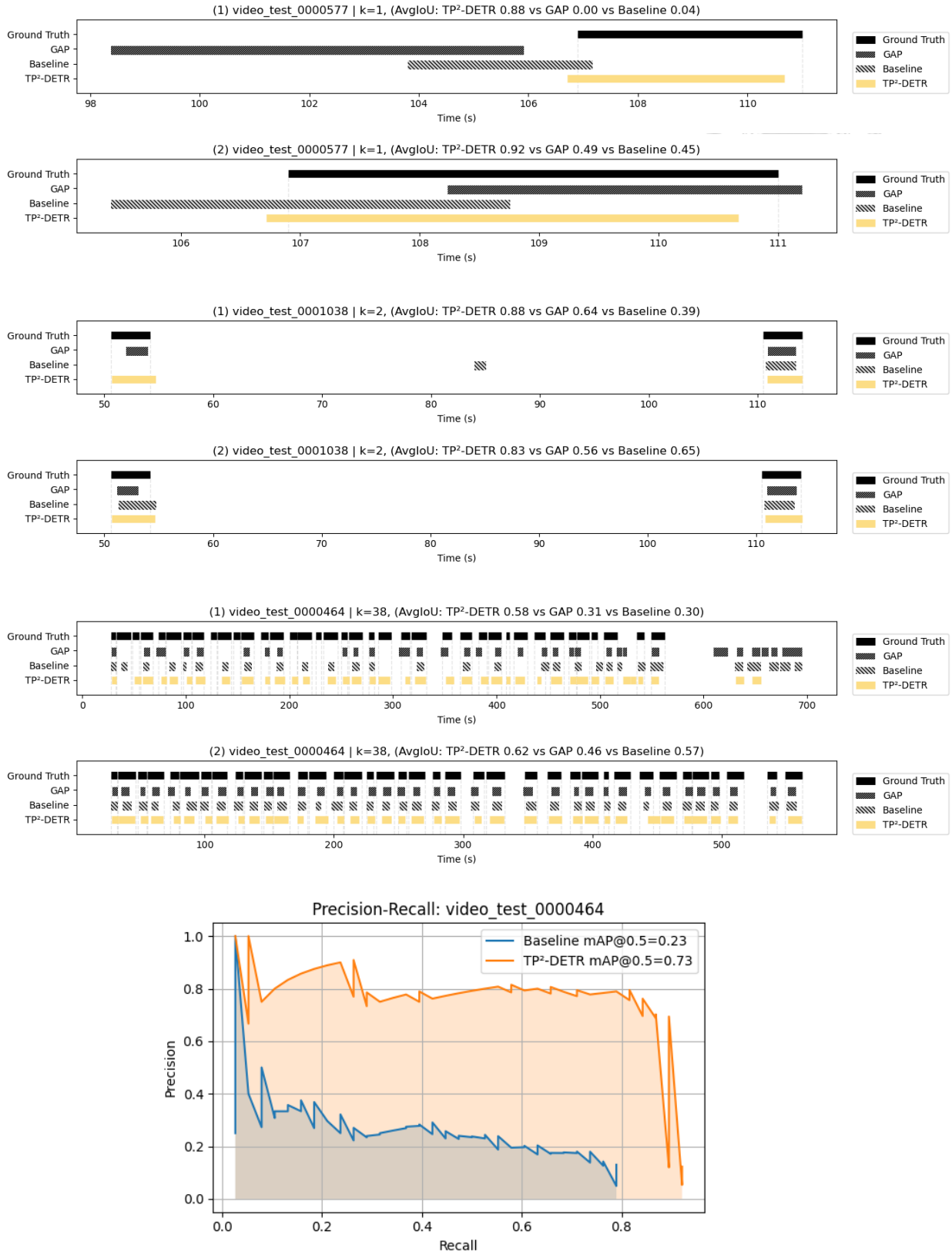
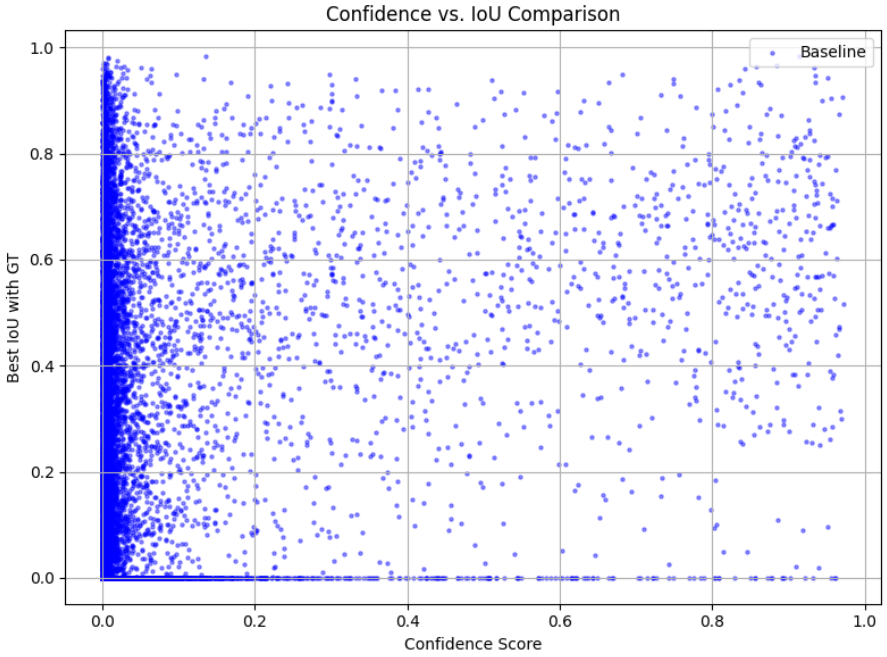
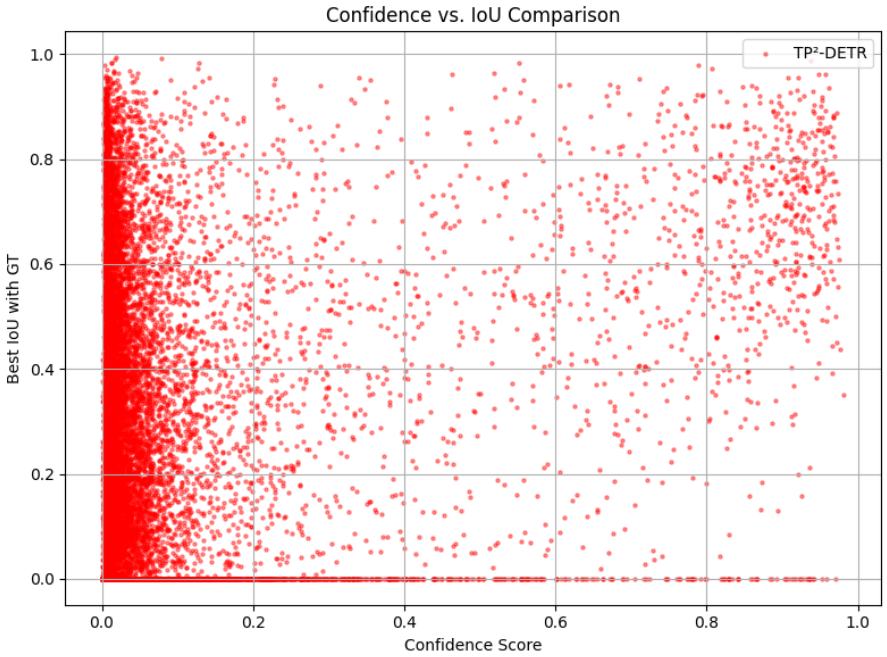


Figure 5.4: **Visualizations of class-agnostic proposals as qualitative results.** We present examples from sparse cases ($k = 1$, $k = 2$) to dense cases ($k = 38$), visualizing the top-k proposals selected in two ways: (1) by actionness scores (following the GAP approach) and (2) by bipartite matching using the same cost as in training. In addition to the intuitive metric AvgIoU for each case, we also provide a precision-recall curve for the dense video (`video_test_0000464`) to illustrate the mAP behavior. All results are from the 50/50 split (split_id = 0) on Thumos14.

doi:10.6342/NTU202503736



(a) Baseline (without temporal FPN)



(b) TP²-DETR (with temporal FPN)

Figure 5.5: **Visualizations of prediction quality (confidence vs. IoU with ground truth).** Each point represents a predicted proposal. TP²-DETR shows a denser concentration in the top-right region, indicating higher-quality proposals in terms of both localization and confidence. Results are reported on Thumos14 using the 50/50 split (split_id = 0).

doi:10.6342/NTU202503736

Chapter 6 Conclusion

In this chapter, we summarize the key contributions of our work and discuss its limitations, along with potential directions for future research.

6.1 Contributions

We propose TP²-DETR, a novel model tailored to fully leverage the strengths of Deformable DETR for the proposal generation subtask in Zero-Shot Temporal Action Localization (ZS-TAL), namely ZS-TAPG. Our main contributions are summarized as follows:

1. **An effective end-to-end framework for ZS-TAPG:** We carefully design and integrate a temporal FPN into Deformable DETR, exploring several architectural variants and finding that the transformer-based design is the most effective. This design enables TP²-DETR to not only take advantage of the sparse and efficient sampling characteristic of deformable attention, but also fully exploit its potential through multi-scale feature integration. Similar to how this combination enhances small object detection in spatial domains, our approach brings analogous benefits to the temporal setting, significantly improving the localization of short actions.

   To ensure training efficiency and model stability in the end-to-end setting, we in-

doi:10.6342/NTU202503736

corporate two additional supervision mechanisms:

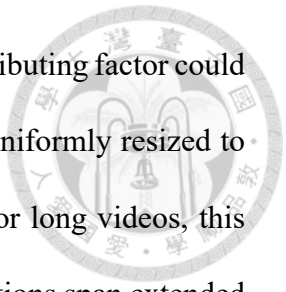
- A multi-scale aware salient head that provides early supervision. Unlike existing CNN-based single-scale designs, our unified MLP-based version is both lightweight and better aligned with the Deformable DETR architecture, supporting multi-scale feature processing without the need for late-stage manual fusion.
- Auxiliary prediction heads placed at intermediate decoder layers, which enable deep supervision and improve convergence. Additionally, the salient head is applied to multiple stages of the network, including outputs from both the temporal FPN and the encoder, providing consistent early supervision throughout the model.

2. **State-of-the-art performance on benchmark datasets:** The experimental results validate the effectiveness of our design, showing that TP²-DETR achieves state-of-the-art performance across most zero-shot split settings on the Thumos14 and ActivityNet1.3 benchmarks. Particularly on Thumos14, which contains a high proportion of short actions, TP²-DETR yields significant average mAP gains of 5.142% and 10.266% under two commonly used zero-shot split settings.

6.2 Limitations and Future Work

Despite the significant improvements TP²-DETR achieves on short-action-dominant datasets, certain limitations remain. In particular, while our model performs well on Thumos14, its results on long-action-dominant datasets such as ActivityNet1.3 are less consistent across different zero-shot split settings. This suggests that the model may not general-

doi:10.6342/NTU202503736



ize as effectively to longer or more complex temporal patterns. A contributing factor could be the preprocessing design in ActivityNet1.3, where each video is uniformly resized to a fixed number of timestamps, regardless of its original duration. For long videos, this may lead to substantial temporal information loss, especially when actions span extended periods. In contrast to approaches that apply sliding window inference to preserve finer temporal granularity, our model processes the entire video as a single sequence, which may limit its capacity to model long-range structures. Future work could explore adaptive resizing schemes or window-based inference strategies to mitigate this issue. Furthermore, enhancing the modeling of long-range dependencies—such as through adaptive temporal scaling or more granular control over the local window sizes in self-attention—may improve the model's robustness across diverse action durations. Our current framework, for instance, does not specifically analyze the impact of window size in the local multi-head self-attention used in the temporal FPN, which may influence its responsiveness to actions of varying lengths.

In addition, our model represents each timestamp as a single token. Inspired by efficient transformer designs such as EdgeViT [3], future work could explore using tokens that cover larger and more context-aware temporal regions. This may help reduce temporal redundancy, enhance local context modeling, and improve efficiency, especially for long videos.

Another limitation of our current design is that it focuses solely on the proposal generation subtask within the ZS-TAL framework, without explicitly addressing the proposal classification component. Future extensions could enrich TP²-DETR by incorporating vision-language alignment or class-aware modules, enabling the model to tackle the full ZS-TAL pipeline in a unified manner, and allow for examining how improvements in

doi:10.6342/NTU202503736

proposal quality may influence classification outcomes. Moreover, the current design follows a two-stage paradigm, where proposal generation and classification are decoupled. Reformulating it as a one-stage method could be a promising direction to reduce the risk of error propagation between stages, as highlighted in prior studies such as STALE [26], and potentially improve both the efficiency and robustness of the overall system.

Additionally, although our salient head is trained independently as an auxiliary component to facilitate early supervision, we further investigate the behavioral alignment between its predictions (i.e., salient logits) and the predicted segments from the primary prediction head (i.e., bounding box head), as shown in Figure 6.1. We observe that salient peaks, particularly those from the encoder output, often align well with the predicted segments, despite the absence of any explicit cross-head connection during training. This implicit consistency suggests the potential for cross-head information sharing, motivating future work to explore mechanisms for shared or guided learning between these two components to further enhance proposal accuracy.

doi:10.6342/NTU202503736

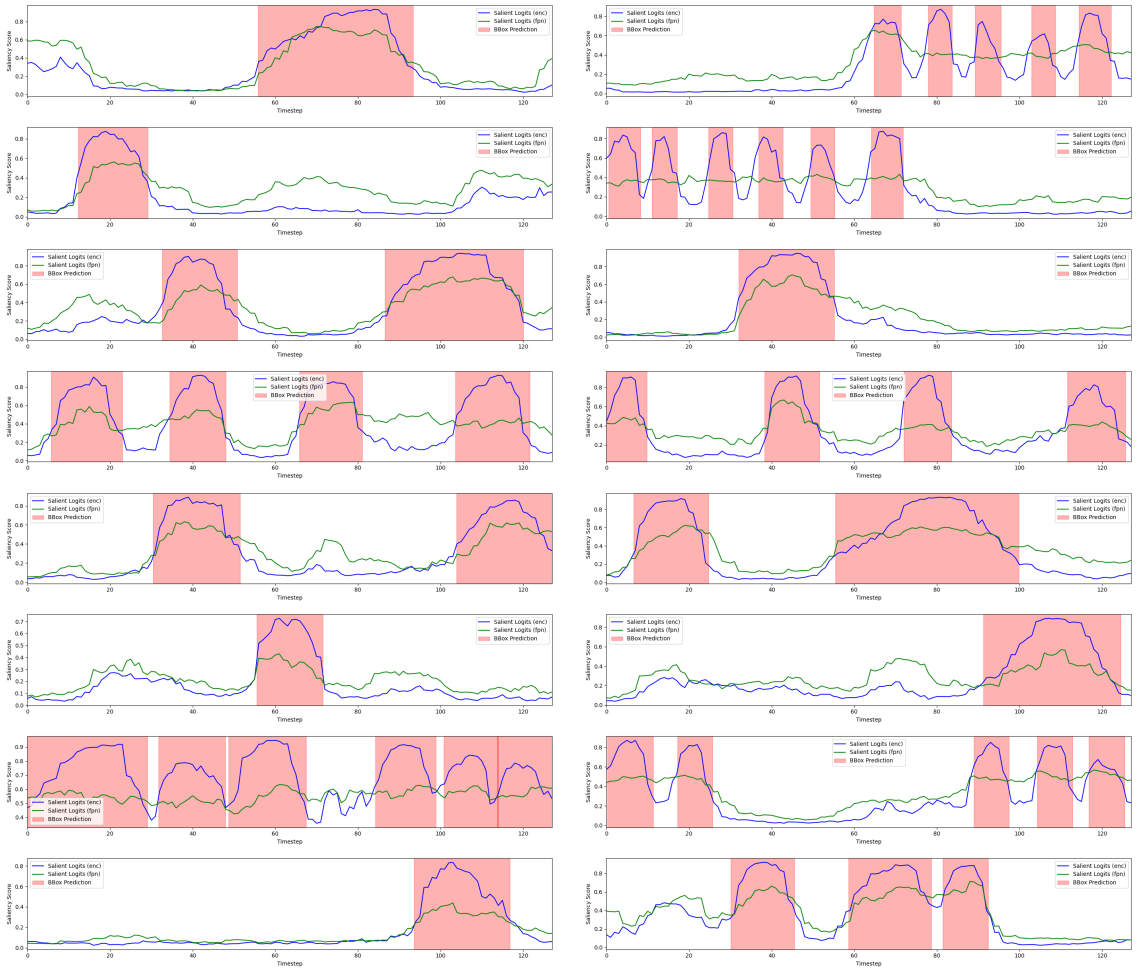


Figure 6.1: **Visualization of salient logits (blue: from encoder, green: from temporal FPN) and predicted proposals (red) from matched queries for a randomly sampled training batch of videos on Thumos14 using the 50/50 split.** Each subfigure shows the predicted saliency scores (y-axis) across timesteps (x-axis). The visualizations show that salient peaks—particularly those from the encoder output—often align with the predicted segments, despite the salient head being trained independently. This supports our hypothesis of potential cross-head consistency, which may be leveraged for future improvements.

doi:10.6342/NTU202503736

References

[1] J. Aklilu, X. Wang, and S. Yeung-Levy. Zero-shot Action Localization via the Confidence of Large Vision-Language Models. arXiv preprint arXiv:2410.14340, 2025.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision (ECCV), pages 213–229. Springer, 2020.

[3] Z. Chen, F. Zhong, Q. Luo, X. Zhang, and Y. Zheng. EdgeViT: Efficient visual modeling for edge computing. In Wireless Algorithms, Systems, and Applications – WASA 2022, volume 13644 of Lecture Notes in Computer Science, pages 393–405, Berlin, Heidelberg, 2022. Springer-Verlag.

[4] J.-R. Du, K.-Y. Lin, J. Meng, and W.-S. Zheng. Towards Completeness: A Generalizable Action Proposal Generator for Zero-Shot Temporal Action Localization. In Proceedings of the 27th International Conference on Pattern Recognition (ICPR), pages 252–267. Springer, 2024.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014.

doi:10.6342/NTU202503736

[6] C. Han, H. Wang, J. Kuang, L. Zhang, and J. Gui. Training-Free Zero-Shot Temporal Action Detection with Vision-Language Models. CoRR, abs/2501.13795, 2025.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 961–970, 2015.

[9] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The THUMOS challenge on action recognition for videos "in the wild". Computer Vision and Image Understanding, 155:1–23, Feb. 2017.

[10] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting Visual-Language Models for Efficient Video Understanding. In Proceedings of the European Conference on Computer Vision (ECCV), pages 105–124. Springer, 2022.

[11] C. Ju, Z. Li, P. Zhao, Y. Zhang, X. Zhang, Q. Tian, Y. Wang, and W. Xie. Multi-modal prompting for low-shot temporal action localization. CoRR, abs/2303.11732, 2023.

[12] H.-J. Kim, J.-H. Hong, H. Kong, and S.-W. Lee. TE-TAD: Towards Full End-to-End Temporal Action Detection via Time-Aligned Coordinate Expression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18837–18846. IEEE, 2024.

[13] J. Kim, M. Lee, C.-H. Cho, J. Lee, and J.-P. Heo. Prediction-Feedback DETR for

doi:10.6342/NTU202503736

Temporal Action Detection. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pages 4266–4274, 2025.

[14] J. Kim, M. Lee, and J.-P. Heo. Self-feedback detr for temporal action detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 10252–10262. IEEE, 2023.

[15] H. W. Kuhn. The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly, 2(1-2):83–97, 1955.

[16] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang. DN-DETR: Accelerate DETR Training by Introducing Query DeNoising. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 46(4):2239–2251, 2024.

[17] B. Liberatori, A. Conti, P. Rota, Y. Wang, and E. Ricci. Test-time zero-shot temporal action localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18720–18729. IEEE, 2024.

[18] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu. Learning Salient Boundary Feature for Anchor-free Temporal Action Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3319–3328. IEEE, 2021.

[19] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 3888–3897, 2019.

[20] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang. DAB-DETR: Dynamic anchor boxes are better queries for detr. In International Conference on Learning Representations (ICLR), 2022.

doi:10.6342/NTU202503736

[21] S. Liu, C.-L. Zhang, C. Zhao, and B. Ghanem. End-to-End Temporal Action Detection with 1B Parameters Across 1000 Frames. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18591–18601, 2024.

[22] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai. End-to-End Temporal Action Detection With Transformer. IEEE Transactions on Image Processing, 31:5427–5441, 2022.

[23] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[24] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang. Conditional DETR for fast training convergence. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 3631–3640. IEEE, 2021.

[25] S. Nag, O. Goldstein, and A. K. Roy-Chowdhury. Semantics Guided Contrastive Learning of Transformers for Zero-shot Temporal Activity Detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 6232–6242. IEEE, 2023.

[26] S. Nag, X. Zhu, Y.-Z. Song, and T. Xiang. Zero-Shot Temporal Action Detection via Vision-Language Prompting. In Proceedings of the European Conference on Computer Vision (ECCV), pages 681–697. Springer, 2022.

[27] OpenAI. GPT-4 Technical Report, 2024.

[28] T. Phan, K. Vo, D. Le, G. Doretto, D. A. Adjeroh, and N. Le. ZEETAD: Adapt-

doi:10.6342/NTU202503736

ing Pretrained Vision-Language Model for Zero-Shot End-to-End Temporal Action Detection. CoRR, abs/2311.00729, 2023.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788. IEEE, 2016.

[31] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, and D. Tao. TriDet: Temporal Action Detection with Relative Boundary Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18857–18866, 2023.

[32] D. Shi, Y. Zhong, Q. Cao, J. Zhang, L. Ma, J. Li, and D. Tao. ReAct: Temporal Action Detection with Relational Queries. In Proceedings of the European Conference on Computer Vision (ECCV), pages 324–344. Springer, 2022.

[33] J. Tan, J. Tang, L. Wang, and G. Wu. Relaxed Transformer Decoders for Direct Action Proposal Generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 13506–13515. IEEE, 2021.

[34] J. Tan, X. Zhao, X. Shi, B. Kang, and L. Wang. PointTAD: Multi-Label Temporal Action Detection with Learnable Query Points. In Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS), page 1111. Curran Associates Inc., 2022.

[35] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem. G-TAD: Sub-Graph Localization for Temporal Action Detection. In Proceedings of the IEEE/CVF Conference

doi:10.6342/NTU202503736

on Computer Vision and Pattern Recognition (CVPR), pages 10153–10162. IEEE, 2020.

[36] S. Yan, X. Xiong, A. Nagrani, A. Arnab, Z. Wang, W. Ge, D. Ross, and C. Schmid. UnLoc: A Unified Framework for Video Localization Tasks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 13577–13587. IEEE, 2023.

[37] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han. Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing, 29:8535 – 8548, 2020.

[38] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. Transactions on Machine Learning Research (TMLR), 2022.

[39] C. Zhang, J. Wu, and Y. Li. ActionFormer: Localizing Moments of Actions with Transformers. In Proceedings of the European Conference on Computer Vision (ECCV), pages 492–510. Springer, 2022.

[40] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In International Conference on Learning Representations (ICLR), 2021. Oral Presentation.

[41] Y. Zhu, G. Zhang, J. Tan, G. Wu, and L. Wang. Dual DETRs for Multi-Label Temporal Action Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18559–18569. IEEE, 2024.

doi:10.6342/NTU202503736