Journal of Science and Technology – Data Science and Artificial Intelligence
Volume 36, Issue 1, October 2026, 018–028

HybridLVLM: Semantic Core-Frame Selection and Two-Stage Chain-of-Thought Reasoning for Continuous Driver Action Recognition

Hiep Bui Hoang[1], Binh Nguyen Thuy[2], Linh Vo Thien[2], Quan Nguyen Hong[3], Frédéric Rayar[4], Lan Le Thi[1,*]

[1]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Ha Noi, Vietnam*
[2]*Faculty of Electrical-Electronic Engineering, University of Transport and Communications, Ha Noi, Vietnam*
[3]*Faculty of Information Technology, Viet-Hung Industrial University, Ha Noi, Vietnam*
[4]*University of Tours, LIFAT (EA6300), 37200, Tours, France*
[*]*Corresponding author email: lan.lethi1@hust.edu.vn*

Abstract

Driver action recognition (DAR) is a core capability of advanced driver assistance systems; however, it remains challenging because distracted driver actions involve small interacting objects, subtle hand movements, and severe self-occlusion inside a confined cabin. Most existing methods are trained on pre-segmented video clips with predefined action labels requiring substantial annotation effort and additional labeled data when extending to new action classes, which limits their scalability to real-world applications. Moreover, they cannot localize actions in an untrimmed video stream. This paper presents HybridLVLM, a training-free framework for continuous driver action recognition in untrimmed videos that leverages large pretrained vision-language models. First, the framework transforms every sampled frame into a natural-language caption, filters the captions using a text-only language model to extract distracted action evidence and core frames for video representation. Second, an open-vocabulary detector then spatially localizes the bounding boxes corresponding to the driver and several task-relevant objects on each core frame. Then, these bounding boxes together with the set of core frames are fed forward to a two-stage Chain-of-Thought (CoT) reasoning model to infer an action class for each core frame based on task-relevant visual evidence. Finally, temporal post-processing is applied to refine and extend frame-level recognition to continuous recognition represented by a predicted action tube. Experiments on the UTCDriverAct dataset show that the HybridLVLM attains a Video-mAP of 52.0These findings point to a promising research direction in video understanding for applications without real-time constraints. Additionally, an ablation study on the input sampling rate is also conducted to investigate the effect of the sampling rate on the overall recognition performance. The results indicate that the optimal sampling density depends on action duration, with higher rates for rapid actions and lower rates for gradual actions.

Keywords: Chain-of-thought reasoning, continuous action recognition, core frame selection, driver action recognition, large vision-language model.

1. Introduction

Traffic accidents remain a critical public health problem worldwide, with driver distraction being a major contributing factor. Consequently, reliable driver action recognition (DAR) has attracted increasing attention as an important approach for understanding driver behavior and improving road safety. By identifying and temporally localizing driver actions, DAR provides valuable information about how drivers interact with the vehicle and the surrounding driving environment. Such information can support a wide range of applications, including distracted driving analysis, driver behavior monitoring, safety assessment, human–vehicle interaction, and post-event analysis. Beyond road safety, DAR also provides a foundation for studying driver behavior patterns and developing more intelligent and human-centered transportation systems.

As a subfield of human activity recognition (HAR), DAR inherits several common challenges, such as background clutter, substantial intra-class variations, and high inter-class similarities. Moreover, DAR also faces several domain-specific challenges arising from the confined space of the vehicle cabin and the small size of task-relevant objects, such as cigarettes, mobile phones, and bottles. These objects are also frequently occluded by the driver's hands or other body parts, making them difficult to detect and recognize. In general, DAR is divided into isolated and continuous action recognition. While isolated action recognition is formulated as a classification problem, where the input is a pre-trimmed video clip of a single driver action. Continuous action recognition deals with untrimmed video streams containing multiple consecutive driver actions and aims to recognize the distracted actions and determine their temporal boundaries. Existing approaches mainly rely on CNN-based methods [1–4] and

p-ISSN 3093-3331
e-ISSN 3093-3323
https://doi.org/10.51316/jst.194.dsai.2026.36.1.3
Received: Aug 26, 2026; Revised: Oct 5, 2026;
Accepted: Oct 9, 2026; Online: Oct 15, 2026

Transformer-based methods [5–8]. Although these approaches have achieved promising results, they typically follow a supervised learning paradigm, requiring labeled training data and assuming a predefined set of action classes.

Recent years have witness a strong transition from traditional models towards large vision-language models (LVLMs). One of the key advantages of LVLMs is that they are pre-trained on large-scale image-text datasets. Therefore, they can provide clear descriptions of the interaction between a person and a surrounding object in natural language. Additionally, they can accommodate new actions by simply modifying the prompt rather than retraining the model. Recent studies have begun to exploit this capability for driver monitoring [9, 10]. However, these works still assume temporally segmented inputs: predictions are aggregated by majority voting inside event boundaries that are given in advance. Applying an LVLM directly to an untrimmed video is not straightforward either, since the number of visual tokens grows linearly with video length and quickly exceeds the context window of the language backbone.

This paper addresses both issues with HybridLVLM, a training-free framework for continuous DAR. The key idea of the proposed method is to move the expensive part of the reasoning task from the visual domain to the language domain. The proposed HybridLVLM framework consists of two main phases. First, core frames are selected by using a large language model (LLM). Second, the selected frames are analyzed through a two-stage Chain-of-Thought (CoT) reasoning process based on textual evidence. The details of the proposed framework are presented in Section 3. Additionally, an ablation study on the input sampling rate is also conducted to investigate the effect of the sampling rate on the overall recognition performance. In summary, the effectiveness of the HybridLVLM framework is primarily attributed to two key components, as follows:

- An action-semantic core-frame selection module that employs a text-only LLM to filter image captions and a dense retriever to identify their corresponding original frames.
- A two-stage CoT reasoning procedure that separates visual evidence extraction from action classification, improving the discrimination of visually similar actions and enabling a more transparent reasoning process.

The rest of this paper is organized as follows. Section 2 discusses briefly some remarkable studies on DAR. The proposed framework will be introduced in Section 3 and Section 4 reports the experimental protocol and results. Section 5 provides some conclusions and outlines the direction for future work.

2. Related work

2.1. Driver Action Recognition

As aforementioned, Convolutional neural networks (CNNs) have long been a traditional approach to DAR task. Nguyen *et al.* [3] proposed an end-to-end framework for continuous recognition that splits the problem into two independent stages: a sliding window produces short segments classified by MoViNet [11] with multi-view voting, and a post-processing extends isolated recognition results to continuous ones. Following a similar sliding-window paradigm, Tran *et al.* [2] adopted X3D-Large [12] as the clip classifier and combined K-fold cross-validation with a three-camera ensemble, followed by a threshold-based temporal localization stage. Other studies emphasize the important role of the interaction between human hands and surrounding objects. Lu *et al.* [1] integrated deformable and dilated convolutions into a Faster R-CNN backbone [13] to detect small objects, while Qamar *et al.* [4] fused an appearance stream with body-pose and human-object interaction streams. Because each clip is processed independently within a short sliding window, these methods struggle to model long-range semantic dependencies between consecutive actions.

Transformer architectures model global spatial relations and have been widely adapted to DAR [5, 7, 8]. Pizarro *et al.* [6] embedded pose estimation inside a video transformer through learnable pose tokens, and Xu *et al.* [14] focus only on informative joints for action representation due to occlusion in the confined space of the vehicle cabin. Addressing the critical aspect of hand-object relations, Bui *et al.* [7] propose to use the holistic interaction transformer (HIT) [15] for DAR task, in which person–object–hand interactions are modeled in both the RGB and pose branches and then merged through an attentive fusion module.

2.2. Vision-Language Models for Driver Behavior Analysis

Hasan *et al.* [9] exploited the zero-shot transfer ability of a contrastive image-text embedding space and proposed four variants of increasing complexity, from direct zero-shot classification to a video-level model. Taking a more structured approach to capture fine-grained actions, Zhang *et al.* [10] proposed the Distracted Driving Language Model, which decomposes each frame into anatomical regions (whole body, head, left and right hand) through a human pose estimator, rephrases the class labels into descriptive sentences, and aggregates a fine-tuned and a zero-shot visual LLM through a reasoning chain. Both works confirm that language supervision transfers well to cabin imagery, but both classify temporally segmented events whose boundaries are known beforehand. The problem of

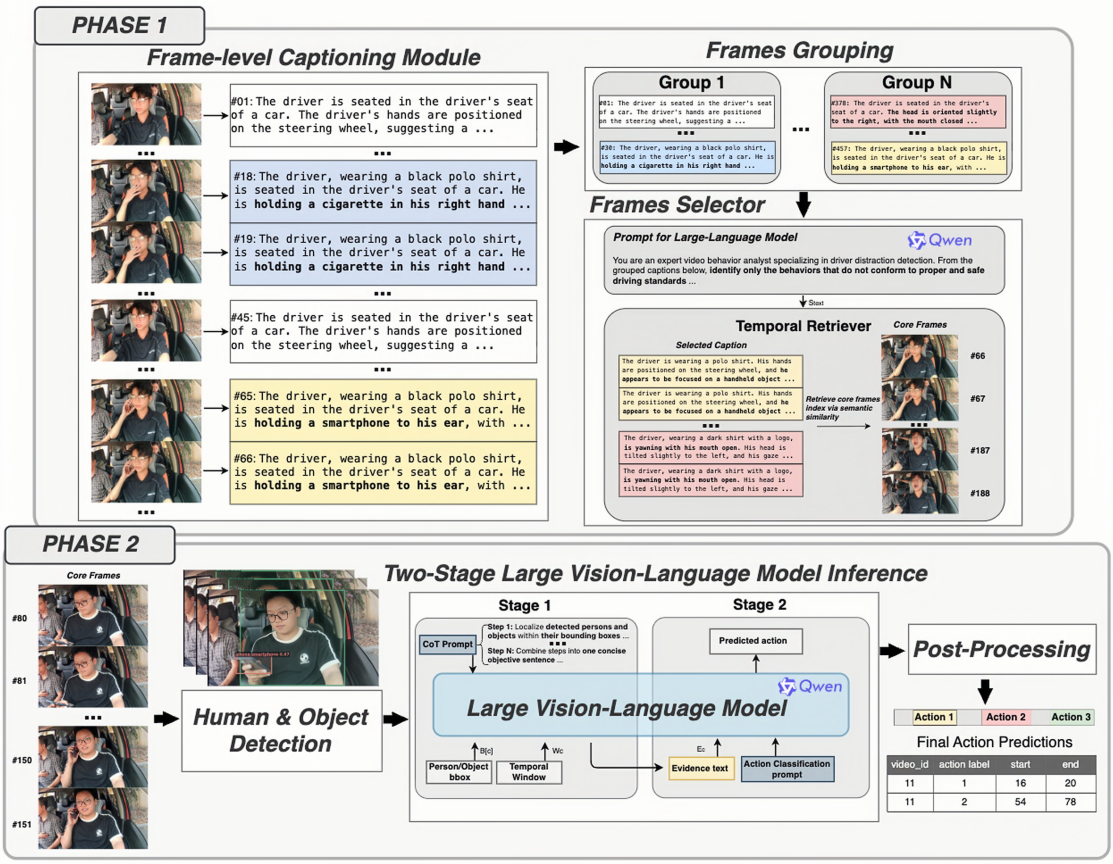

Fig. 1. Overview of the proposed HybridLVLM framework for continuous driver action recognition in driver behavior analysis system

jointly detecting and localizing actions in an untrimmed stream, which is the setting considered here, remains open.

Based on the above analysis, this work introduces a training-free framework HybridLVLM combines the two ideas for untrimmed video: core frame selection is question-agnostic and driven by driver action semantics rather than by visual clustering, and reasoning chain leverages both spatial and temporal cues. Details of the proposed framework will be introduced in the following section.

3. Proposed Method

3.1. Problem Formulation and Overview

It is important to emphasize that this study focuses on the continuous recognition task, whose primary objective is to generate a set of spatio-temporal action tubes, as defined by the following equation:

$$\mathscr{T} = \{(c, B_{\text{action}}, A_c, S_c)\} \tag{1}$$

where c denotes the index of the frame, B_{action} is the bounding box corresponding to an interested driver action, A_c and S_c are the predicted action label and its confidence score. The proposed framework for continuous DAR is illustrated in Fig. 1, consisting of four crucial steps: (1) sampling and core-frame selection, (2) driver and task-relevant object detection, (3) CoT reasoning, (4) post-processing. The main goal of the first step is to identify and select the most informative frames from the raw input video, referred to as core frames $\mathscr{F}_{\text{core}}$, which serve as inputs to the action recognition process. For each core frame, an open-vocabulary detector localizes the driver and task-relevant objects (e.g., phones, cigarettes, and bottles), providing the detected bounding boxes $\mathscr{B}_c$. These detection results are subsequently fed into a two-stage CoT reasoning model to infer the action class associated with each core frame. Finally, the frame-level predictions are integrated through several post-processing strategies, including linking, smoothing, and merging, to produce the final action prediction for the input video and construct the corresponding spatio-temporal action tube. It is worth noting that the proposed framework follows a training-free approach, in which none of the deep learning models require additional training or fine-tuning.

Each model in the framework is selected according to its role, computational cost, and compatibility with the other stages. In the first step, all sampled frames are passed to a vision-language model to generate the corresponding caption. Qwen2-VL is chosen for this step due to its strong visual understanding and ability to generate concise descriptions of driver posture, hand

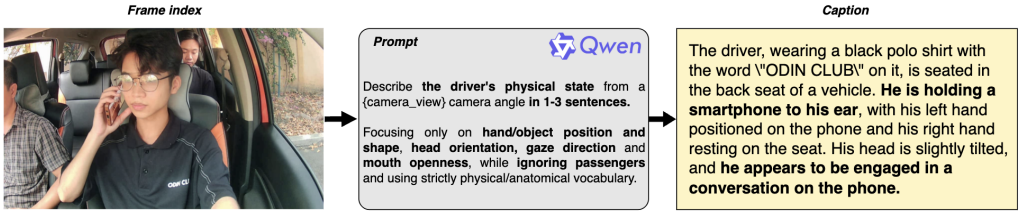

Fig. 2. Prompt design and an example of the frame-level caption generated for a sampled frame

positions, and task-relevant objects. The core frames are then extracted from the captions using Qwen2.5-Instruct. This phase operates on text only, therefore, a language model is sufficient and avoids the memory and latency overhead of a vision encoder. The CoT reasoning phase is the most computationally demanding, as it processes multiple images, localizes the driver, and generates structured outputs. Using models from the same Qwen family ensures consistent interfaces across phases and simplifies data flow. Each component can be replaced with a stronger model when additional computational resources are available.

3.2. Sampling and Behavior-Semantic Core-Frame Selection

In the first step, a raw input video is sampled to generate a sequence of frames $\mathscr{F}$, from which a set of core frames $\mathscr{F}_{\text{core}}$ are selected for video representation. The primary objective of this technique is to eliminate redundant information, which helps accelerate processing and reduce memory usage. This process is performed through three sequential sub-steps as below.

3.2.1. Frame-level captioning

Each frame $f_i \in \mathscr{F}$ is fed forward to Qwen2-VL [16] model together with a scene-description prompt to create its corresponding caption, as formulated in Eq.(2):

$$\mathscr{C}[i] = \text{Qwen2-VL}\big(f_i, \texttt{prompt_caption}\big), \quad \forall f_i \in \mathscr{F} \tag{2}$$

The results of this sub-step is to produce a set of caption $\mathscr{C} = \{\mathscr{C}[i]\}_{i=1}^{|\mathscr{F}|}$, in which every element is a natural-language sentence describing the visual content of the corresponding frame. Qwen2-VL integrates a Vision Transformer (ViT) encoder with a decoder-only language model. For image captioning task, the superior performance of Qwen2-VL can be attributed to its ability to generate context-faithful captions while maintaining an acceptable computational cost across all sampled frames. Fig. 2 illustrates the prompt design and a representative caption. Mapping visual information to the language domain enables the retrieval and recognition processes to operate on compact textual representations, which is more efficient than directly processing visual features. In this stage, we use the following prompt for image captioning task: *"You are an objective video analyst observing a driver from a {camera_view} camera angle. Describe the driver's physical state in 1-3 natural sentences"*. This prompt focuses on the positions of both hands, the physical shapes of task-relevant objects, and their spatial relationships with the driver's body.

3.2.2. Caption clustering and behavior filtering

In this sub-step, the above image captions are clustered into consecutive context blocks of size g, shown as the following equation:

$$\mathbb{G} = \text{Cluster}(\mathscr{C}, g) \tag{3}$$

It is worth noting that the clustering process is based on the principle that the consecutive image captions describing the same visual content, with no significant differences between them, are grouped into a single cluster. Reasoning over caption blocks rather than individual captions allows the downstream language model to maintain sufficient continuous context for recognizing driver actions spanning multiple frames while improving computational efficiency. After caption clustering, each caption block $G \in \mathbb{G}$ is then passed to Qwen2.5-Instruct model [17] with a behavior-filtering prompt that retains only the captions describing unsafe driving:

$$\begin{aligned} \mathscr{S}_G &= \text{Qwen2.5}\big(G, \texttt{prompt_filter}\big) \\ \mathscr{S}_{\text{text}} &= \bigcup_{G \in \mathbb{G}} \mathscr{S}_G \end{aligned} \tag{4}$$

For this task, we use a behavior-filtering prompt as follows: *You are an expert video behavior analyst specializing in driver distraction detection. From the list of frame captions below, select ONLY the captions that clearly describe unsafe or distracted driving behavior.* To extract a distracted action corresponding to an image caption, we also provide definitions of normal driving or distracted driving. *A standard safe posture is defined strictly as: both hands resting on the steering wheel, head and gaze oriented forward towards the road, eyes open, and mouth closed.* Distracted driving is defined as any behavior that differs from the baseline driving condition described above.

The primary goal of using a text-only model at this stage is to reduce computational cost by avoiding the requirement to encode all video frames and their corresponding captions. More importantly, filtering based on behavioral semantics rather than matching

a predefined query makes the module query-agnostic, which is essential for detecting multiple behavior classes within a single untrimmed video. The obtained result of this stage is a set of filtered sentences $\mathscr{S}_{\text{text}}$ which focuses only on unsafe driver action. However, the connection between each filtered sentence and the original frame is lost because of the above process (clustering, filtering). This issue is further addressed in the following sub-step, as detailed below.

3.2.3. Temporal information retrieval

The crucial task of this sub-step is to identify the original frame which is the most suitable to a filtered-sentence s in the set $\mathscr{S}_{\text{text}}$. For this, a filtered sentence s and an original caption $\mathscr{C}[j]$ are fed forward to the input of BGE-M3 model [18] to generate two textual embeddings $\mathbf{e}_s = \text{BGE}(s)$ and $\mathbf{e}_j = \text{BGE}(\mathscr{C}[j])$, respectively. BGE-M3 is chosen thanks to its ability to capture semantic similarity between two short sentences rather than relying on word-level matching. The frame whose caption is semantically closest to s is recovered based on the cosine similarity between $\mathbf{e}_s$ and $\mathbf{e}_j$, shown as in the following equation:

$$\text{sim}_j = \frac{\mathbf{e}_s \cdot \mathbf{e}_j}{\|\mathbf{e}_s\| \, \|\mathbf{e}_j\|}, \qquad idx^* = \arg\max_j \text{sim}_j \tag{5}$$

The retrieved frames are then ordered by their temporal positions, with repeated frames removed, as multiple filtered captions may correspond to the same frame, yielding the final core-frame set $\mathscr{F}_{\text{core}}$.

A simpler option is to let the LLM return the indices of the selected captions. However, Phase 1 uses a lightweight LLM to keep core-frame selection inexpensive. Tiny LLMs are prone to hallucination, especially when asked to output numerical identifiers. They may return indices that are shifted or do not exist in the input. A wrong index selects the wrong frame, and this error cannot be detected afterwards. We therefore ask the LLM to output caption text instead. BGE-M3 then maps each output sentence in S_{text} to its most similar original caption $C[j]$ and thus to its frame index. This mapping is based on semantic similarity rather than exact matching. It therefore remains correct even when the generated sentence has minor errors such as extra prefixes or missing words.

3.3. Open-Vocabulary Person and Object Detection

In the second step, an open-vocabulary detector is used to identify regions of interest, focusing on the driver and task-relevant objects, within a core image in the set $\mathscr{F}_{\text{core}}$. In this study, Grounding DINO model [19] is chosen as it offers a flexible and extensible object vocabulary, enabling new task-relevant objects to be incorporated by appending noun phrases to without retraining the detector.

In our method, Grounding DINO model is queried twice for each core frame with two different textual prompts to generate spatial bounding boxes, described as below:

$$B_{\text{obj}} = \text{GDINO}(f_c, P_{\text{obj}}, \theta_{\text{obj}}) \tag{6}$$

$$B_{\text{person}} = \text{GDINO}(f_c, \text{"person"}, \theta_{\text{person}}) \tag{7}$$

where P_{obj} is a predefined list covering the driver's body parts (`driver face`, `driver hand`) and objects associated with distracted driving (`phone`, `smartphone`, `bottle`, `cup`, `can`, `cigarette`). The prompts used for Groudning DINO are shown as below:
GDINO_OBJECT_PROMPT = "driver face . driver hand . phone . smartphone . bottle . cup . cigarette"
GDINO_PERSON_PROMPT = "person"

The two box sets are merged and filtered by an anchor-distance criterion followed by non-maximum suppression, giving the spatial buffer $\mathscr{B}_c$. To filter out unsuitable boxes, we use some thresholds including box matching threshold, box matching threshold, and the number of filtered box, denoted as τ_{q}^{box}, τ_{q}^{text}, and K_q, where $q \in \{\text{obj}, \text{person}\}$. For each predicted box, Grounding DINO computes a similarity score with every token in the prompt. The two thresholds τ_{q}^{box} and τ_{q}^{text} are used to determine which predicted boxes are kept. Accordingly, all predicted boxes with similarity scores below these thresholds are discarded and only K_q bounding boxes with the highest scores are retained. In all our experiments, these thresholds are set as follows: $\tau_{\text{q}}^{box} = \tau_{\text{q}}^{text} = 0.25$ and $K_q = 3$. The low threshold improves recall for small or partially occluded objects such as cigarettes and hand-held phones, while limiting the number of boxes to three per query reduces irrelevant detections before the filtering step.

3.4. Two-Stage Chain-of-Thought Reasoning

The primary target of this step is to predict a driver action class for each core frame in the set $\mathscr{F}_{\text{core}}$ based on the spatial bounding boxes $\mathscr{B}_c$ extracted from the previous step. To address this task, these two sets $\mathscr{F}_{\text{core}}$ and $\mathscr{B}_c$ are passed to a two-stage CoT model Qwen2.5-VL [**Bai2025Qwen25VLTR**] which decomposes driver action recognition task into two sequential stages, called evidence extraction and decision making. In the first stage, Qwen 2.5-VL model attempts to extract task-relevant visual evidence, such as the driver and interacting objects. These evidences are subsequently exploited in the second stage to infer the action class presented in the examined core frame. In other words, the main principle of two-stage CoT technique is to separate *what is observed* from *what the observation means*, making the reasoning process more transparent by explicitly providing the visual evidence used for action recognition. For the first stage of CoT reasoning, we create the following prompt: *You are analyzing {n_frames} consecutive frames from a {camera_view}*

camera. Frame {anchor_idx} of {n_frames} is the ANCHOR frame to classify. It is worth noting that this prompt focuses on the position and movement of hands and objects across frames. Additionally, mouth state and lip movements are also considered. And the prompt: *"Classify the driver's activity based strictly on the provided physical synthesis"* is utilized in the second stage to classify the target driver actions.

In reality, a driver action usually takes place over a period of time corresponding to multiple consecutive frames. Therefore, in this work, a temporal window of width w centered on each core frame f_c is used to provide sufficient information for evidence extraction in the first stage. The output of the first stage consists of a set bounding boxes presented the distracted driver actions along with their corresponding textual evidence, denoted as E_c and B_{action}. The second stage classifies the action using only the core frame f_c and the evidence E_c produced in the first stage, as shown in the following equation:

$$A_c, S_c = \text{Qwen2.5-VL}\big(f_c, E_c, \texttt{prompt}_2\big) \tag{8}$$

The evidence sentence acts an information bridge between the two stages: he classifier inherits the temporal context captured in the first stage without re-encoding the entire temporal window, leading to a lower computational cost. It is important to emphasize that two-stage CoT is the recognition result at frame-level, known as a raw prediction $\mathscr{I}_{\text{raw}}$ consisting of a set of core frames and their corresponding predicted driver action classes. To achieve continuous recognition, this raw prediction will be refined through several post-processing strategies, discussed in more detail below.

3.5. Temporal Post-Processing

This temporal post-processing step aims to extend the frame-level recognition obtained in the previous step to the continuous recognition. For this, the proposed framework follows several strategies introduced in the work of Nguyen et al. [3], including linking, removing, and merging. Firstly, all core frames sharing the same label are linked together to form an action tube which consists of some different action segments. It is worth noting that core frames are sparsely sampled, therefore the boundary core frame index does not accurately indicate when an action transition occurs. The temporal boundary of each action segment is defined as the midpoint between the last core frame of the preceding segment and the first core frame of the current segment. Moreover, if the temporal gap between two action segments which are assigned the same action label is smaller than $\Delta t = 5$s they will be merged to prevent a single continuous action from being fragmented into several small segments. Finally, linear interpolation generates the temporally sparse boxes B_{action} between adjacent core frames, producing the final set $\mathscr{T}$

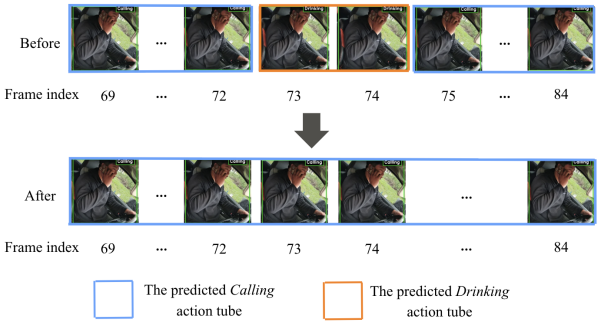

Fig. 3. An example for the effectiveness of the post-processing step, the recognition result is smoother

with a continuous box sequence. Fig. 3 illustrates the effectiveness of post-processing step in provide the final results for continuous action recognition, where noise is removed and the predicted action tube smoother and longer.

4. Experimental Results and Discussion

4.1. Dataset

To show the effectiveness of the proposed framework in driver action recognition task, we have conducted some experiments on UTCDriverAct dataset [7, 8]. For data collection process, 18 volunteers are required to perform sequentially six distracted actions, including *Smoking*, *Texting*, *Calling*, *Drinking*, *Yawning*, and *Talking to Passenger*. This dataset consists of total 40 videos captured by two static cameras mounted at front and rear of the vehicle. All videos are recorded at 30 frames per second (fps) and last approximately four to five minutes. The first 24 videos from 10 subjects are used for training, while the remaining 16 videos from 8 subjects are used for testing. It is important to noting that the HybridLVLM framework is training-free and requires no model training, all results reported below are obtained on the 16 testing videos, which is the same partition used for action recognition evaluation in other existing methods [7, 8]. Fig. 4 shows some sample frames in UTCDriverAct dataset with six driver actions of interest.

4.2. Evaluation Metrics

As aforementioned, continuous recognition not only identifies the action being performed but also determines its temporal boundaries, including the start and end time of each target action.

In this study, we use three popular metrics that are Spatial-Temporal Tube IoU (STT-IoU), Video mean Average Precision (Video-mAP), and F1-score are used. **(STT-IoU)** [20] computes the overlap between a predicted tube and a ground-truth tube across both spatial and temporal dimensions.

$$\text{STT-IoU} = \left(\frac{\text{Temporal Intersection}}{\text{Temporal Union}}\right) \times \text{Average IoU}_{2D} \tag{9}$$

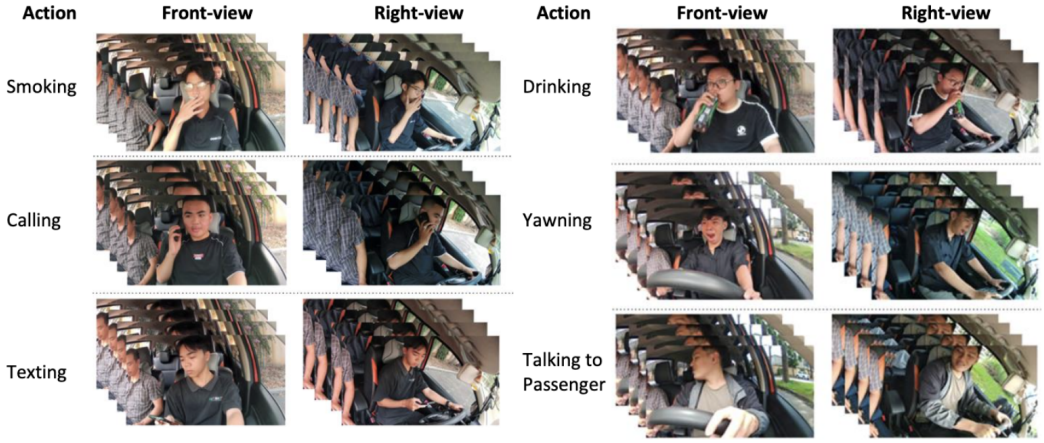

Fig. 4. The six driving actions annotated in the UTCDriverAct dataset [7]

This metric ensures that a high score is obtained only when the prediction is accurate in both the spatial and temporal domains.

Video-mAP treats a predicted tube as a true positive when its STT-IoU with the corresponding ground-truth tube exceeds a pre-defined threshold. Unlike image-level mAP, it evaluates complete tubes and therefore reflects classification and spatio-temporal localization simultaneously.

$$\text{Video-mAP} = \frac{1}{C}\sum_{c=1}^{C} AP_c \tag{10}$$

F1-score@k evaluates the effectiveness of the proposed framework at a single operating point through the harmonic mean of precision and recall, where a predicted action tube is considered as a true positive only if its STT-IoU exceeds the threshold k:

$$\text{F1-score@}k = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

4.3. Experimental Results

All experiments run on Ubuntu 20.04.6 LTS with a single NVIDIA GeForce RTX 3060 GPU with 12 GB of VRAM. Parameters used in all experiments include a sampled rate of $r = 1$fps, a context block size of $g = 15$, and the temporal window in the first CoT stage of $w = 5$ frames. The merging tolerance in post-processing is $\Delta t = 5$s. Additionally, all model weights are the publicly released checkpoints and no deep-learning model is fine-tuned on UTCDriverAct dataset.

4.3.1. Per-class recognition performance evaluation

Table 1 reports the per-class results of HybridLVLM at STT-IoU = 0.3. Averaged over the six action classes, HybridLVLM framework reaches an F1-score of 61.4% and a Video-mAP of 52.0% without any training on driving data. Across all action classes, the proposed framework achieves the highest performance on *Calling* class with an F1-score of 82.1% driven by a perfect Recall of 100.0% (no False negative) at a Precision of 69.6%. This action involves a large, clearly visible object and a distinctive hand-to-ear posture, both of which are reliably described by the captioning module and consistently retained by the action filter.

Conversely, *Texting* class obtains the lowest F1-score (40.6%) despite the second-highest Recall (87.5%), because its Precision collapses to 26.4% with 39 False positives, several times more than any other class. The framework therefore over-predicts *Texting*, most likely due to confusion with driver actions involving a downward gaze and a similar hand position, such as *Calling*. However, *Yawning* class exhibits the opposite tendency: it attains the highest Precision of all classes (72.7%) but only 50.0% Recall, with as many False negatives (24) as True positives. This can be explained by the fact that yawning is the shortest action, therefore, discriminative cues might be lost when sampling at one frame per second.

Talking to Passenger class yields equal Precision and Recall at 50.0%, a balanced but moderate result for an action that lacks any interacting object and is defined mainly by head orientation. Across the six classes, average Recall (72.5%) clearly exceeds average Precision (58.3%). This asymmetry follows directly from the design of the core-frame selection module, which is deliberately biased toward retaining captions containing even subtle indications of driver distraction, leading to an increase in the number of False positives. For a driver monitoring system, in which a missed warning is more costly than a spurious one, this operating point is the appropriate trade-off. Fig. 5 shows an example of recognition results when applying HybridLVLM framework on testing video with ID 01_018_01_0_front in UTCDriverAct dataset. As shown in this figure, *Calling* class is incorrectly classified as *Texting* class. This can be explained that these two actions exhibit similar driver postures and interactions with a mobile phone, making these two activities difficult to distinguish.

Table 1. Per-class recognition performance of HybridLVLM on the UTCDriverAct dataset at an STT-IoU threshold of 0.3. The best value per metric is in bold and the second best is underlined

Actions	TP	FP	FN	Precision (%)	Recall (%)	F1-score@0.3 (%)	AP@0.3 (%)
Smoking	28	11	18	71.8	60.9	65.9	56.6
Calling	16	7	0	69.6	**100.0**	**82.1**	**68.3**
Texting	14	39	2	26.4	87.5	40.6	49.1
Drinking	19	13	3	59.4	86.4	70.4	57.1
Yawning	24	9	24	**72.7**	50.0	59.3	41.4
Talking to Passenger	12	12	12	50.0	50.0	50.0	39.5
All classes	–	–	–	58.3	72.5	61.4	52.0

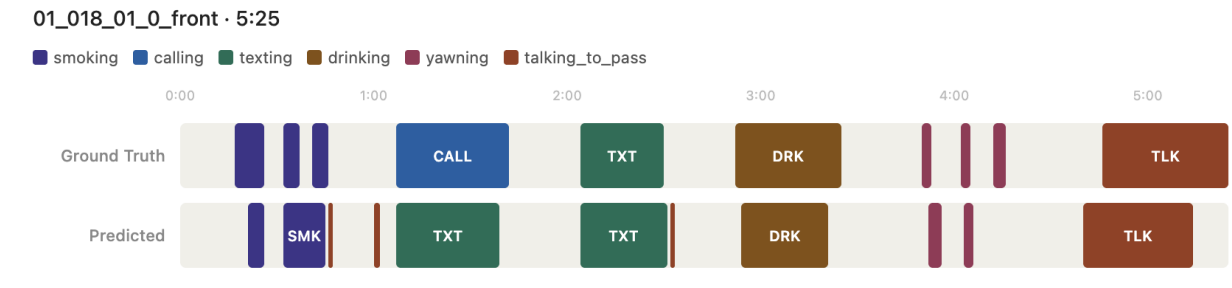

Fig. 5. An example image for continuous driver action recognition on testing video with ID 01_018_01_0_front in UTCDriverAct dataset

Table 2. Inference time and memory consumption per frame of each step in HybridLVLM framework

Computational cost		Inference time (s/frame)	Memory usage (GB)
Phase 1	Frame captioning	1.35	9.28
	Grouping & core frame selection	0.45	6.65
Phase 2	Human/Object detection	0.48	2.2
	CoT reasoning	6.32	9.6
	Classification	4.65	7.28
Total		13.65	

Furthermore, we evaluate the computational cost of all steps in HybridLVLM framework, including frame captioning, grouping and core-frame selection, human/object detection, CoT reasoning, and classification. To this end, we apply the proposed framework on video 01_012_01_0_front in the testing set of UTCDriverAct. Table 2 reports the inference time and memory usage measured for each frame. In phase-1, the inference time and memory usage are 1.35 s and 9.28 GB for frame captioning, and these values are 0.45 s and 6.65 GB for grouping and core-frame selection step. In phase-2, the human/object detection module requires only 0.48 s and 2.2 GB of memory for inference. While the two-stage CoT incur inference times of 6.32 s and 4.65 s, respectively, with memory consumption of 9.60 GB and 7.28 GB. Overall, the two-stage CoT requires approximately 10.97 s to determine the action class of each core frame, accounting for 80.36% of the total processing time of the end-to-end framework. This computational cost makes HybridLVLM more appropriate for practical applications that do not require real-time processing such as driver behavior analysis systems.

4.3.2. Ablation study of HybridLVLM components

Table 3. Ablation results obtained by removing individual HybridLVLM components

Phase 1	Phase 2		Video-mAP (%)	F1-score (%)
Core-frame selection	CoT reasoning	Prediction		
✓	✓	✓	50.92	61.54
	✓	✓	48.26	57.14
✓		✓	44.68	53.85

To evaluate the role of HybridLVLM components in DAR task, we conduct some additional experiments. In the first scenario, core-frame selection stage is removed, an input video is sampled at the frame rate of 1 fps and these extracted frames are passed to the second stage to identify the target driver actions and their temporal boundaries. In contrast, two-stage CoT reasoning module is removed in the second scenario, where core frames are chosen from the first stage and fed into Grounding DINO detector to generate bounding boxes corresponding to target objects. These bounding boxes are then used as

Table 4. Ablation on the input sampling rate at STT-IoU = 0.3. For each driver action class, the better of the two configurations is shown in bold on the F1-score and Video-mAP columns

Actions	**Sampling rate**	**TP**	**FP**	**FN**	**Precision (%)**	**Recall (%)**	**F1-score@0.3 (%)**	**AP@0.3 (%)**
Smoking	$r = 1$	28	11	18	71.8	60.9	**65.9**	56.6
	$r = 3$	26	8	21	76.5	55.3	64.2	**58.2**
Calling	$r = 1$	16	7	0	69.6	100.0	**82.1**	**68.3**
	$r = 3$	15	6	5	71.4	75.0	73.2	60.1
Texting	$r = 1$	14	39	2	26.4	87.5	40.6	49.1
	$r = 3$	17	22	2	43.6	89.5	**58.6**	**55.9**
Drinking	$r = 1$	19	13	3	59.4	86.4	**70.4**	57.1
	$r = 3$	17	15	1	53.1	94.4	68.0	**61.0**
Yawning	$r = 1$	24	9	24	72.7	50.0	**59.3**	41.4
	$r = 3$	26	7	37	78.8	41.3	54.2	**42.0**
Talking to Passenger	$r = 1$	12	12	12	50.0	50.0	**50.0**	**39.5**
	$r = 3$	10	14	9	41.7	52.6	46.5	25.5
Average	$r = 1$	–	–	–	58.3	72.5	**61.4**	**52.0**
	$r = 3$	–	–	–	60.9	68.0	60.8	50.5

the input to an LVLM model to infer directly which driver action is being performed. Table 3 indicates the obtained results when removing each component of HybridLVLM framework. It is worth noting that these results are computed on a representative video in the testing set of UTCDriverAct dataset, identified by video ID 01_012_01_0_front. As can be seen in Table 3, when removing the first phase in HybridLVLM framework, video-mAP and F1-score are reduced by 2.66% and 4.40%, respectively, compared to those in the full framework. A more substantial reduction is observed when the first stage of the CoT reasoning module is removed, resulting in a 6.24% decrease in Video-mAP and a 7.69% decrease in F1-score. These confirm the contribution of each component to the proposed framework, particularly CoT reasoning module.

4.3.3. Ablation study on the input sampling rate

In real-world driving scenarios, driver actions exhibit substantial differences in duration and movement characteristics. Some actions are short and rapidly changing, whereas others last for longer periods with little movement. To investigate the effect of the sampling rate on the overall performance of the proposed framework, we also conduct some additional experiments with different sampling rates, $r = 1$ and $r = 3$. It is important to note that these experiments are also performed with an STT-IoU threshold of 0.3. The obtained results are indicated in Table 4.

The benefit of denser sampling is particularly clear for driver actions characterized by distinctive hand gestures. For *Texting* class, AP@0.3 rises from 49.1% to 55.9% and F1-score@0.3 from 40.6% to 58.6%, with False positives falling from 39 to 22; *Drinking* likewise gains in AP@0.3, from 57.1% to 61.0%, and reaches a Recall of 94.4%, the highest value in the table. Denser sampling lets the captioning module capture more frames in the short interval during which these gestures unfold, providing richer textual information for filtering and retrieval. However, driver actions involving little postural change follow the opposite trend. The Recall of *Yawning* drops from 50.0% to 41.3% and its False negatives grow from 24 to 37, while AP@0.3 stays essentially unchanged. *Talking to Passenger* degrades on both metrics, with AP@0.3 falling by almost one third, from 39.5% to 25.5%. Averaged across the six action classes, $r = 1$ therefore remains the better configuration (Video-mAP 52.0% against 50.5%), even though $r = 3$ is preferable for three individual classes in terms of AP.

Based on the above analysis, we conclude that denser sampling is more beneficial for recognizing actions with rapid movements than those with little movement. This phenomenon can be explained by two mechanism in the core-frame selection module. On the one hand, increasing the sampling rate generate more captions per unit of time and hence finer descriptions for fast driver actions, resulting in the gains on *Texting* and *Drinking* classes. On the other hand, the size of each caption block is set to $g = 15$ which does not depend on the sampling rate, therefore, the real-time duration covered by a context block is reduced from about 15s (at $r = 1$) to 5s (at $r = 3$). As a result, the filtering model may observe only a fragment of a long-duration action (e.g, *Drinking* or *Calling*), leading to recognition failures or the selection of an unrepresentative frame.

4.3.4. Comparison with a supervised interaction transformer

Table 5 presents a comparison between HybridLVLM framework and HIT4DAR [7], a supervised learning method. HIT4DAR framework is considered as a hybrid architecture of CNN and Transformer, where 3D-CNN is used to extract video features and a spatial transformer encodes the driver

Table 5. Comparison between the supervised HIT4DAR baseline and the training-free HybridLVLM on UTCDriverAct at STT-IoU = 0.5. The better value per behavior and metric is in bold

behavior	**F1-score@0.5 (%)**		**AP@0.5 (%)**	
	HIT4DAR	**HybridLVLM**	**HIT4DAR**	**HybridLVLM**
Smoking	46.8	**49.4**	33.6	**41.5**
Calling	**71.0**	66.7	**56.0**	54.0
Texting	**87.5**	40.1	**78.9**	48.1
Drinking	**59.1**	54.0	42.6	**47.6**
Yawning	**40.9**	32.9	**26.8**	23.1
Talking to Passenger	**61.2**	22.5	**48.9**	27.0
Average	**61.1**	44.3	**47.8**	40.2

pose. By this way, both visual and skeleton data are exploited for driver action representation. Furthermore, a cross-attention module is introduced to model the interactions between the driver's hands and task-relevant objects for action recognition. It is important to emphasize that HIT4DAR framework is trained on the training set of UTCDriverAct. Both methods are evaluated under an identical protocol at STT-IoU = 0.5.

Overall, the supervised baseline retains an advantage: HIT4DAR reaches a Video-mAP of 47.8% against 40.2% for HybridLVLM, a gap of 7.6 percentage points, and wins on four of the six behaviors for both metrics. The margin is widest on *Texting* (F1-score 87.5% against 40.1%) and *Talking to Passenger* (F1-score 61.2% against 22.5%). This reflects the benefit of training directly on the target distribution: HIT4DAR learns behavior-specific person–object interaction patterns and temporal boundaries from labeled data, whereas HybridLVLM relies entirely on the generic reasoning ability of the underlying language models and consequently confuses behaviors whose appearance is close. However, supervised methods require labeled data, which can require substantial time and effort to annotate. They also require retraining when applied to a new dataset and cannot recognize previously unseen actions. HybridLVLM addresses these limitations as a training-free framework that supports flexible open-vocabulary driver action recognition without task-specific retraining.

5. Conclusion

This paper introduced HybridLVLM, a training-free framework for continuous driver action recognition on untrimmed video. Experiments are conducted on UTCDriverAct dataset show that HybridLVLM framework achieves a Video-mAP of 52.0% and a mean F1-score of 61.4% at STT-IoU = 0.3 without any training on driving data. These results suggest a promising research direction for driver behavior analysis applications that do not require real-time processing.

Despite its promising performance, HybridLVLM relies on a multi-stage pipeline in which errors in caption generation, core-frame selection, or object detection may propagate to subsequent reasoning stages. Moreover, its temporal modeling relies on fixed sampling and context parameters together with heuristic post-processing. Future work will therefore investigate joint visual-semantic core-frame selection and adaptive temporal reasoning to reduce error propagation and enable more accurate localization of action boundaries in untrimmed videos. Confidence-aware reasoning across different stages will also be explored to improve the robustness of the overall framework.

Acknowledgment

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-23-1-4064.

References

[1] M. Lu, Y. Hu, and X. Lu, Driver action recognition using deformable and dilated Faster R-CNN with optimized region proposals, Applied Intelligence, vol. 50, no. 4, pp. 1100–1111, Apr. 2020. https://doi.org/10.1007/s10489-019-01603-4

[2] M. T. Tran, M. Q. Vu, N. D. Hoang, and K.-H. N. Bui, An effective temporal localization method with multi-view 3D action recognition for untrimmed naturalistic driving videos, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Jun. 2022, pp. 3168–3173.

[3] H.-Q. Nguyen, T.-B. Nguyen, T. K. Tran, V.-N. Hoang, T.-L. Le, T.-H. Tran, and H. Vu, End-to-end deep learning-based framework for driver action recognition, in 2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), 2022, pp. 1–6.

[4] H. I. Qamar, U. Saeed, and M. Hussain, Driver distraction detection using a multi-stream deep fusion network, in International Conference on Computing and Emerging Technologies, 2023, pp. 97–104.

[5] Y. Ma, L. Yuan, A. Abdelraouf, K. Han, R. Gupta, Z. Li, and Z. Wang, M2DAR: Multi-view multi-scale driver action recognition with vision transformer, pp. 5287–5294, Jun. 2023. https://doi.org/10.1109/CVPRW59228.2023.00557

[6] R. Pizarro, R. Valle, L. M. Bergasa, J. M. Buenaposada, and L. Baumela, Pose-guided multi-task video transformer for driver action recognition, arXiv preprint arXiv:2407.13750, Jul. 2024.

[7] H.-H. Bui, K.-H. Bui, T.-L. Vo, H.-Q. Nguyen, T.-B. Nguyen, T.-T. Dao, and T.-L. Le, HIT4DAR: Holistic interaction transformer for driver action recognition, Transport and Communications Science Journal, vol. 77, iss. 4, pp. 500–514, May 2026. https://doi.org/10.47869/tcsj.77.4.12

[8] Q.-T. Nguyen, T.-B. Nguyen, H.-Q. Nguyen, and T.-L. Le, MAE4DAR: Masked autoEncoder for driver action recognition, VNU Journal of Science: Computer Science and Communication Engineering, Jun. 2026. https://doi.org/10.25073/2588-1086/vnucsce.7153

[9] M. Z. Hasan, J. Chen, J. Wang, M. S. Rahman, A. Joshi, S. Velipasalar, C. Hegde, A. Sharma, and S. Sarkar, Vision-language models can identify distracted driver behavior from naturalistic videos, IEEE Transactions on Intelligent Transportation Systems, vol. 25, iss. 9, pp. 11602–11616, Sep. 2024. https://doi.org/10.1109/TITS.2024.3381175

[10] K. Zhang, S. Wang, N. Jia, L. Zhao, C. Han, and L. Li, Integrating visual large language model and reasoning chain for driver behavior analysis and risk assessment, Accident Analysis & Prevention, vol. 198, Feb. 2024, Art. no, 107497. https://doi.org/10.1016/j.aap.2024.107497

[11] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, MoViNets: Mobile video networks for efficient video recognition, pp. 16015–16025, Mar. 2021. https://doi.org/10.48550/arXiv.2103.11511

[12] C. Feichtenhofer, X3D: Expanding Architectures for Efficient Video Recognition, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 200–210, 2020.

[13] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in Advances in Neural Information Processing Systems, pp. 91–99, Dec. 2015.

[14] Z. Xu and J. Xu, Spatiotemporal decoupling attention transformer for 3D skeleton-based driver action recognition, Complex & Intelligent Systems, vol. 11, pp. 1–12, Feb. 2025, Art. no. 195.

[15] G. J. Faure, M.-H. Chen, and S.-H. Lai, Holistic interaction transformer network for action detection, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 3329–3339.

[16] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, *et al.*, Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, arXiv preprint arXiv:2409.12191, 2024.

[17] S. S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, Qwen2.5 technical report, 2025, [Online] URL: https://arxiv.org/abs/2502.13923.

[18] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, arXiv preprint arXiv:2402.03216, 2025.

[19] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection, in European Conference on Computer Vision, 2024, pp. 38–55.

[20] C. Feichtenhofer, A. Pinz, and R. P. Wildes, Spatiotemporal residual networks for video action recognition, in Advances in Neural Information Processing Systems, 2016, pp. 3468–3476.