Università degli Studi di Padova

Dipartmento di Matematica "Tullio Levi-Civita"

Corso di Dottorato in Computer Science
Brain, Mind and Computer Science

Ciclo XXXVI

Human Action Anticipation: Deep Learning Approaches Across Diverse Domains

Supervisore:
Prof. Lamberto Ballan

Dottorando:
Nada Salah Mahmoud Osman

2024

To the selfless woman, who dedicated her life for us
To my mother, Soheir El-Shamandy

To my dear father, Prof. Salah Osman
To my beloved husband, Ahmed
To my sister, Noha, and brothers, Muhammad and Amr

Acknowledgements

Alhamdulillah, I praise and thank **Allah**, the Almighty, the Most Gracious, and the Most Merciful, for granting His blessings upon me throughout my acdemic journey, and for giving me strength and courage to successfully complete this work.

First and foremost, I wish to express my sincereset gratitude to my supervisor, **Prof. Lamberto Ballan**, for his unwavering support and guidance throughout this research journey. My gratitude also extends to all my colleagues at the Visual Intelligence and Machine Perception (VIMP), especially **Guglielmo Camporese** and **Pasquale Coscia**, for their invaluable collaboration. I am also deeply thankful to **Prof. Marwan Torki** for his support and guidance during my research visit to Alexandria University.

I express special thanks to my husband, **Ahmed**, for his continuous, day-to-day, encouragement, support, and help throughout my work. I also express my warmest thanks to my **father**, **mother**, **sister**, and **brothers** for their endless care and support; their prayers for me have sustained me so far.

January 2024
Nada Osman

Contents

List of Figures

List of Tables

Abstract

Human action anticipation holds fundamental importance across various domains and applications. Anticipating human actions enables proactive decision-making, enhancing efficiency, safety, and overall performance of many systems, including robotic assistance systems, advanced surveillance systems, and autonomous driving, where self-driving cars should be able to anticipate pedestrians' intentions and actions to guarantee people's safety. In this dissertation, our primary focus centers on anticipating human actions within two important domains, representing the two main formulation of action anticipation: 1) Egocentric anticipation of in-kitchen activities; 2) Third-person anticipation of pedestrian actions. However, our research extends to cover the anticipation of the collective behavior patterns in traffic flows, and we extended even further to tackle the domain of abnormal behaviors decoding and recognition.

Firstly, we address the anticipation of in-kitchen activities in first-person inputs, investigating the capability of the anticipation models in adapting to the variable progressing time of the human actions. Some actions happen faster or slower than others, depending on the actor or the surrounding context, which could vary each time and lead to different predictions. Based on this idea, we build upon a well-known action anticipation model, Rolling-Unrolling LSTM (RULSTM), which is specifically designed for anticipating human actions, and propose a novel attention-based technique to simultaneously evaluate the slow and fast features extracted from three different modalities, namely RGB, optical flow, and extracted objects. Two branches process information at different time scales, i.e., frame rates, and several fusion schemes are considered to improve anticipation accuracy. In this regard, we perform extensive experiments on EPIC-KITCHENS-55 and EGTEA Gaze+ datasets and demonstrate that our technique systematically improves the results of RULSTM architecture for Top-5 accuracy metric at different anticipation times.

Then, we move to our second anticipation domain of pedestrian action anticipation in urban scenarios, using third-person input videos. Due to the complex urban environments with many human-human or human-vehicle interactions, anticipating a pedestrian action relies on several clues. This challenging task is often tackled using visual and non-visual features to anticipate future actions from 2 s to 1 s earlier to the event. Our work primarily aims to revise this standard evaluation protocol to forecast crossing events as early as possible. To this end, we conceive a solution upon RULSTM, proposing to envision future features or modalities, that can better infer human intentions using a properly attention-based fusion mechanism. We validate our model against JAAD and PIE datasets and demonstrate that an intent prediction model can benefit from these additional clues for anticipating pedestrian crossing events.

Continuing with pedestrians' action anticipation, we propose a novel approach based on a multi-modal transformer. Our model encodes past observations and produces multiple predictions at different anticipation times. Moreover, we propose to learn the attention masks of our transformer-based model (TAMFORMER: **T**emporal **A**daptive **M**ask Trans**FORMER**) in order to weigh differently present and past temporal dependencies. We investigate our method on several public benchmarks for early intention prediction, improving the prediction performances at different anticipation times compared to the previous works.

Building upon TAMFORMER, we propose and investigate the effect of taking advantage of a language modality in pedestrian action anticipation. We study various captioning techniques of the observed frames, integrating the generated text into our TAMFORMER model. Additionally, we expand the binary crossing/not crossing pedestrian action anticipation into multi-action anticipation. We validate our techniques on a novel large-scale dataset (LOKI), proving the notable effectiveness of including text in increasing the model comprehension and, consequently, increasing the performance.

Transitioning to traffic flow anticipation, we aim to forecast collective behaviors embodied in the traffic flow conditions on a city scale. We introduce a model for anticipating traffic speed, utilizing attention-based spatiotemporal encoding and a dual-graph road-network representation. The dual-graph framework combines spatial and contextual sub-graphs, facilitating the exploration of non-Euclidean spatial correlations and potential contextual similarities within road networks. To dynamically capture spatiotemporal correlation, we employ multi-head self-attention modules capable of discerning temporal and spatial correlations. Additionally, we present a fast conditional diffusion model for spatiotemporal traffic data imputation, employing a high-order pseudo-numerical solver. Through experimentation on two publicly available real-world traffic datasets, our proposed model achieves superior performance compared to existing baselines.

Finally, our exploration extends to the detection of abnormal actions in surveillance settings. Given the difficulty imposed in manually supervising and detecting anomalous events in a vast amount of surveillance videos, the task mainly relies on semi-supervised learning. We introduce a transformer-based temporal-hierarchical model that weighs the impact of the observed actions in classifying a video as anomalous. Implementing a divide-and-conquer approach over the temporal axis, the video is hierarchically segmented into multiple instances, creating distinct temporal patches. Obtaining sub-predictions from these diverse patches enhances the model's ability to estimate abnormality scores within video segments. The initial evaluation of our methodology on a large-scale surveillance dataset gives promising insights into the viability of the proposed approach.

Chapter 1

Introduction

We live in an envisioned future where artificial intelligence is integrated into our daily lives, undertaking many tasks, from trivial to very complicated ones, working and performing side-to-side with humans (Figure 1.1). Robotic assistance systems have become increasingly available, offering physical support in diverse settings, including personalized aid at homes [1], cooking assistance in kitchens [2, 3], or even specialized aids to technicians [4]. As we stroll the roads, we can see an autonomous vehicle with a self-driving system effortlessly navigate the streets [5]. Given this evolving reality, advancing AI models that can decode and anticipate human behavior and actions is a must. Consequently, human action recognition and anticipation emerge as a crucial topic in computer vision research. An assistive robotic platform needs to recognize and anticipate human movements to perform its tasks correctly and safely; similarly, advanced video-surveillance systems [6] require anticipating/decoding human actions and motion to provide timely assistance promptly; and indeed, a self-driving car must be able to anticipate pedestrians' and human drivers' actions to make proper and safe driving decisions [7, 8].

(A) At home assistance [1]

(B) In kitchen assistance [9]

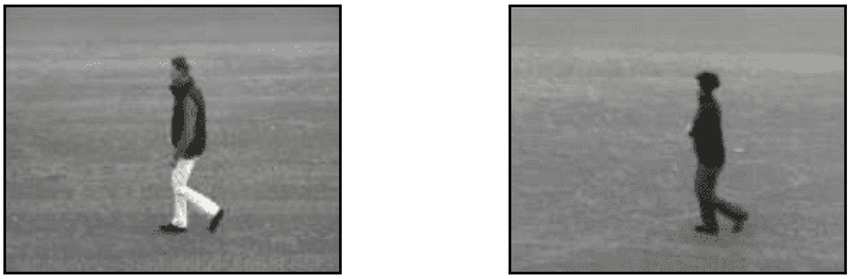
(C) Technical assistance [4]

FIGURE 1.1: Collaboration between AI and humans in various applications

Human behavior decoding and foreseeing take many forms and shapes, including action recognition [10], early action recognition [11], movement and trajectory prediction [12], and, most recently, action anticipation [13]. Our research primarily focuses on action anticipation. The subsequent section outlines the specifics of our task, ascribing its distinctions from other human actions decoding tasks. Following that, we provide insights into the targeted applications of our work.

1.1 Action Recognition and Anticipation Tasks

An action is a sequence of dynamic events and steps that unfold over time, leading to doing or achieving a task or a goal. Time is the central axis in defining an action; therefore, attempting to distinguish an action from a single frame is almost impossible. As illustrated in Figure 1.2, the two frames are from two different activities, yet both are practically identical. Consequently, tasks related to action recognition and anticipation necessitate the analysis of multiple frames, forming a coherent video sequence, catching the temporal relationships, and leading to the identification of the underlying activity.



(A) Walking

(B) Jogging

FIGURE 1.2: Similar instantaneous activities, yet different actions [14]

In **action recognition** (depicted in Figure 1.3) [15,16,16–18,18–23], the model analyzes a sequence of frames, aiming to identify and categorize the action being performed within the observed frames. The model is not required to make predictions; rather, it should proficiently decode and interpret the observed action. The same applies to the branched task of **anomaly/abnormal actions recognition** [24–28], where the task is generalized to predict the type of behavior instead of the specific activity (refer to Chapter 8).

Shifting to **early action recognition** (as shown in Figure 1.4) [29–40], the objective transforms to distinguishing the unfolding action by analyzing only the early frames of the overall activity, presenting an additional layer of complexity.

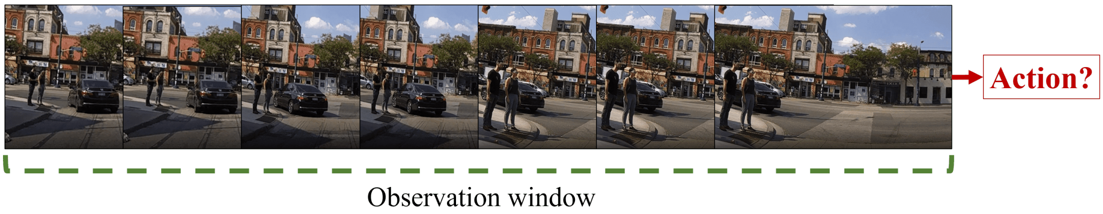


FIGURE 1.3: **Action Recognition:** Recognize actions within a given sequence of frames.

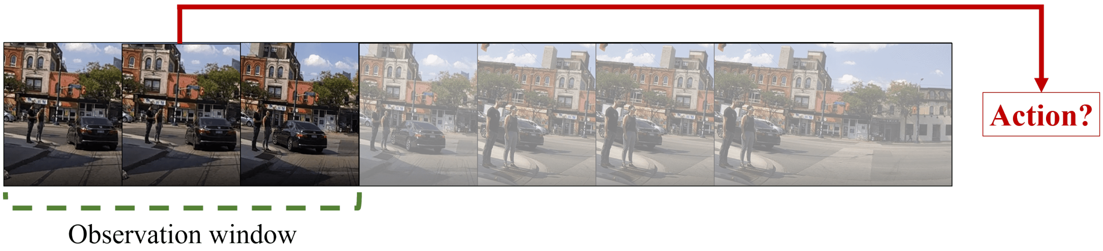


FIGURE 1.4: **Early Action Recognition:** Distinguish actions from the initial frames.

On the Other hand, **Action prediction/anticipation** introduces a future prediction. After observing a historical sequence of frames, the model attempts to forecast future events. The predicted event could be as early as the subsequent event immediately following the conclusion of the observation [41], as depicted in Figure 1.5, or advancing further to our task, where the model is challenged not to predict the immediate subsequent action but rather to anticipate the action that will happen after a specified duration in the future, denoted as the anticipation time t_a [42] (refer to Figure 1.6). Throughout the rest of the dissertation, we will refer to this task as **action anticipation**.

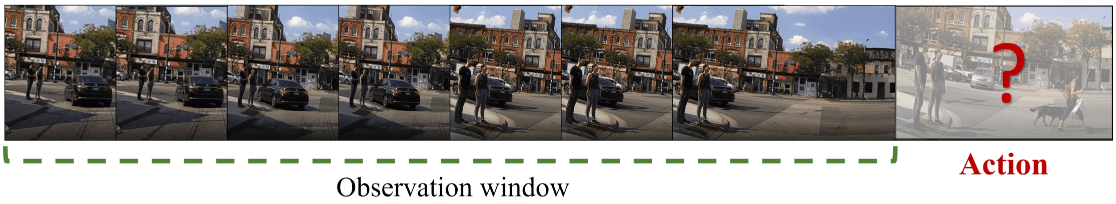


FIGURE 1.5: **Action Prediction:** Forecast the next event following the observation.

Specifically, the model observes a set of historical frames f_{obs}, extracts diverse features and modalities representing both the human subject and the contextual environment, and then predicts the event occurring after t_a seconds in the future. It is noteworthy that the choice of t_a varies depending on the specific application under consideration.

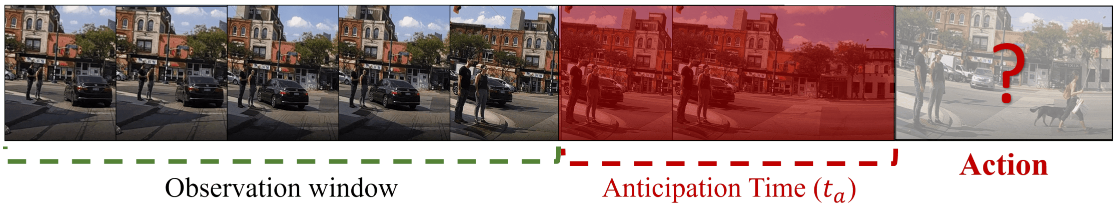


Figure 1.6: **Action Anticipation:** Anticipate actions occurring after a determined time in the future.

Figure 1.7 illustrates our anticipation protocol, where the model starts with a warm-up encoding sequence of frames S_{enc}, followed by the anticipation sequence S_{ant} representing the decoding stage, and producing predictions at different anticipation times. The decoding starts with the earliest anticipation at $t_a = t_s - \alpha \times T$ seconds, where t_s is the time at which the future action occurs, T is the number of anticipation steps, and α is the timestep between subsequent anticipations. Consequently, the latest anticipation is at $t_a = t_s - \alpha$ seconds. Again, the choice of t_a, represented by t_s and α, depends on the considered application.

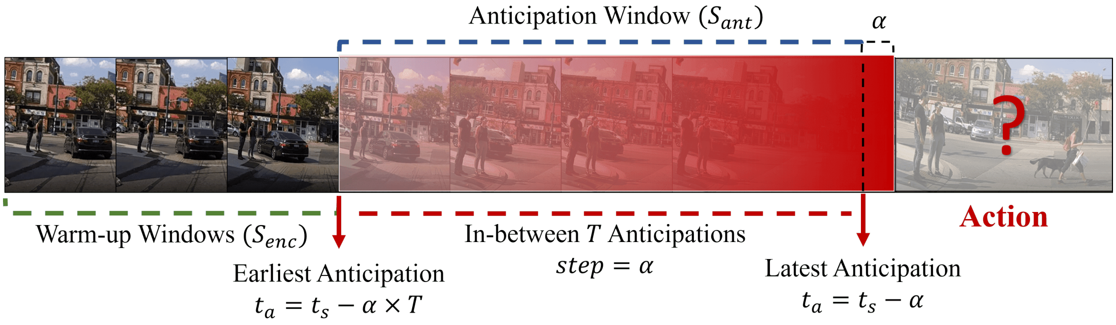

Anticipation Window (S_{ant})
α
?
Action
Warm-up Windows (S_{enc})
In-between T Anticipations
$step = \alpha$
Earliest Anticipation
$t_a = t_s - \alpha \times T$
Latest Anticipation
$t_a = t_s - \alpha$

Figure 1.7: Our anticipation protocol

1.2 Selected Research Domains

As highlighted earlier, action recognition and anticipation play vital roles in diverse applications, requiring the establishment of many benchmarks for evaluation. Our research focuses on two main anticipation domains, representing the two main formulation of the action anticipation task: In-Kitchen activities anticipation in egocentric settings, and urban-scenarios pedestrian actions anticipation in third-person settings. However, we also expand the anticipation task to the domain of traffic flow forecasting within city-scale networks. As an additional domain, our research shifts focus from anticipation to action recognition, specifically in the context of detecting anomaly actions in surveillance environments. To comprehensively address our chosen research domains, we will now elaborate on the benchmarks employed within each domain.

1.2.1 In-Kitchen Activities Anticipation

As illustrated in Figure 1.1b, robotic assistance has made significant strides inside modern kitchens for cooking and cleaning activities. However, the kitchen environment is tremendously complicated, incurring a massive number of possible actions and activities. Accordingly, the research community works on providing reliable benchmarks for evaluating action recognition and anticipation in the kitchen environment. We work with two popular datasets of kitchen activities: EpicKitchens-55 [43] and EGTEA Gaze+ [44], where Table 1.1 reports the details of the two datasets. Both datasets are egocentric videos where participants capture their daily activities in the kitchen with a mounted camera on their heads (Figure 1.8).

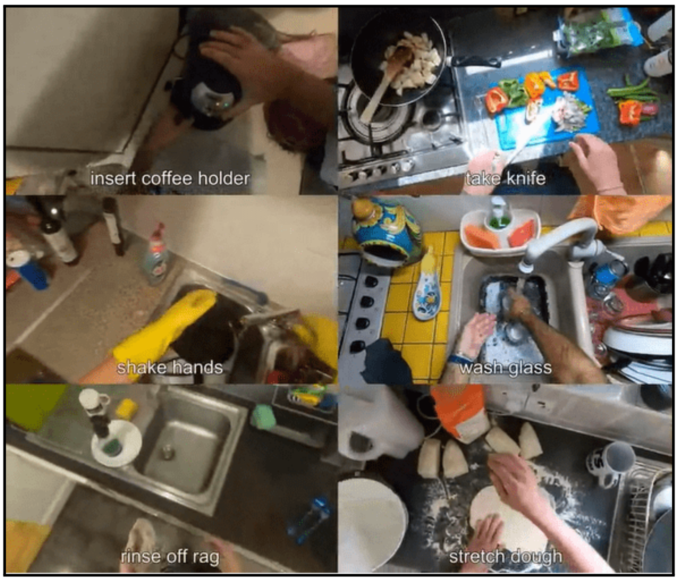


Figure 1.8: Egocentric EPIC-KITCHENS dataset [45]

1.2.2 Pedestrians Action Anticipation

The anticipation of pedestrian actions presents a more subtle challenge. While the range of possible pedestrian actions in street scenarios is limited, the context is notably complex, encompassing a wide range of external influences and more critical actions (e.g., crossing or not crossing the street). Additionally, it is harder to interpret and comprehend the behaviors and the goals of humans enter and exit

	EPIC-KITCHENS-55 [43]	EGTEA Gaze+ [44]
Dataset Length	55 hours	28 hours
Number of Participants	32	32
Actions Verbs	125 verb	19 verb
Objects Nouns	352 noun	51 noun
Unique (verb, noun) Action Pairs	2, 513 actions	106 actions

Table 1.1: EPIC-KITCHENS-55 VS EGTEA Gaze+

the view in third-person videos, compared to the focused egocentric videos, due to the absence of a unified thread of actions and objects interactions. Consequently, considerable interest is in anticipating human actions in urban scenarios, paired with the generation of datasets that capture urban scenes, enabling the evaluation of street-related tasks, including pedestrian action prediction and anticipation. We work with three well-known urban-scenes datasets, captured with a camera mounted on an ego-vehicle (Figure 1.9): JAAD [7], PIE [8], and LOKI [46]. Table 1.2 presents an overview of these datasets.

Figure 1.9: Urban scenarios datasets [47]

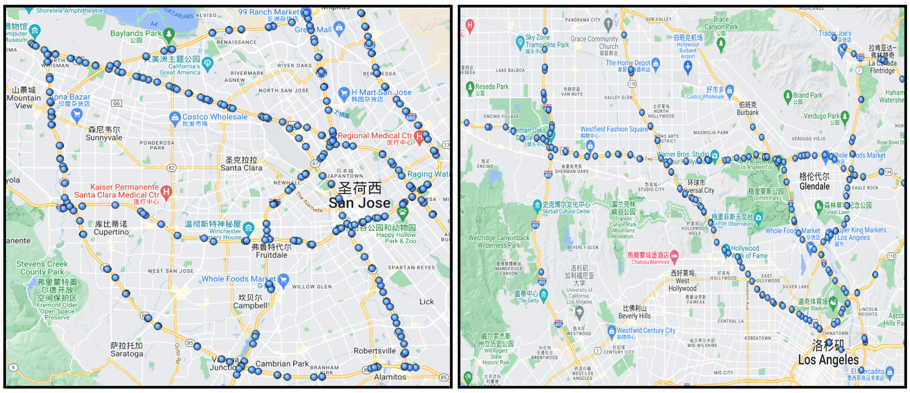
1.2.3 Traffic Flow Anticipation

In the same context of urban-environment anticipations, forecasting traffic flow conditions on road maps represents another aspect of intelligent systems related to urban environments. This has prompted the research community to establish evaluation benchmarks for this particular problem. Differing from the standard action anticipation task, which analyzes video frames to provide short-term per-second anticipations, traffic conditions anticipation analyzes traffic signals, such as traffic speed, within road maps to offer hourly predictions. In our approach to this

	JAAD [7]	PIE [8]	LOKI [46]
Annotation Frame rate	30 FPS	30 FPS	5 FPS
Number of Pedestrians	2793	1842	9226
Behavioural Annotations	686	1842	9226
Actions	Crossing Not Crossing	Crossing Not Crossing	Moving Stopping Waiting to Cross Crossing Other
Ego-Vehicle Information	-	Speed GPS coordinates Heading direction	GPS coordinates Heading direction
Pedestrian Attributes	-	Age - Gender	Age - Gender

Table 1.2: Urban scenarios datasets

task, we rely on two real-world traffic flow datasets: METR-LA and PEMS-BAY [48]. Figures 1.10a and 1.10b illustrate the sensor networks, capturing the traffic flow signals, and road maps of the two datasets, while Table 1.3 provides a brief description of both datasets.



(a) METR-LA (b) PEMS-BAY

Figure 1.10: Road map and sensor network of traffic flow datasets

1.2.4 Anomaly Actions Recognition

Deviating to the domain of anomaly recognition in surveillance videos, advanced surveillance systems are now integral to nearly all security setups, where the identification and detection of abnormal behaviors that may pose security threats is a central task. Consequently, there is a growing focus on security-related tasks and

	METR-LA	PEMS-BAY
Traffic Flow Sensors	207	325
Map Edges (Road Connections)	1515	2369
Timestamps	34272	52116
Time Span	01/03/2012 - 27/06/2012	01/01/2017 - 30/06/2017
Time Interval	5 minutes	
Daily Hours	00:00 - 24:00	

Table 1.3: Statistics of Traffic flow datasets

benchmarks specifically tailored for surveillance videos. Extensive benchmarks have been established based on the vast publicly available surveillance videos. In this context, we use a large-scale benchmark dataset of anomaly actions, namely UCF-Crime [49] (Figure 1.11), containing 13 different anomaly actions within 1900 videos, summing-up to 128 hours. The dataset provides per-video action labels but not per-frame action labels, where the exact location of the anomalous behavior in the video is unknown. Therefore, it is considered to be a semi-labeled dataset.

Figure 1.11: UCF-Crime dataset [47]

1.3 Research Objectives

As highlighted in the preceding sections, human action decoding and anticipation hold significant importance within the field of computer vision, given its broad applications. In our research, we pick two recent and vital domains of human action anticipation: Firstly, the anticipation of in-kitchen activities, a field gaining widespread attention, especially with recent advancements in the EPIC-KITCHENS [43] benchmark. Secondly, our exploration extends to predicting pedestrians' actions and intentions, which is particularly crucial in the context of self-driving vehicles, denoting a turning point in combining safety and efficiency in autonomous driving. Our primary research objective is to explore these two anticipation domains. Expanding our research scope further, we aim to enrich our anticipation and action decoding investigations by incorporating additional tasks. One such task involves the anticipation of traffic flow conditions, encompassing the analysis of traffic patterns and holding significance in intelligent transportation systems. Another additional task focuses on decoding normal and abnormal behaviors in surveillance videos, a critical aspect in advanced security systems.

Our research objectives revolve around designing, developing, and implementing deep learning and computer vision approaches within our chosen action decoding and anticipation domains. We seek to investigate the capabilities of these approaches in interpreting and understanding human actions, ultimately aiming to advance the research field in the selected domains. This advancement is realized through improved task performance and the provision of novel insights into addressing the challenges, exploring various methodologies and techniques.

To outline the primary objectives of our research, we aim to:

1. Explore and understand the capabilities of deep learning and computer vision models in decoding and recognizing human behavior, ultimately leading to the modeling of action anticipation across diverse environments.

2. Develop innovative methodologies and techniques to enhance the performance of human action decoding and anticipation in various applications and settings.

3. Investigate the efficacy of different feature extraction and modeling techniques in extracting valuable behavioral and contextual information from observed data, videos, frames, or signals.

4. Examine the influence of time, observation/anticipation time, and interval time on the interpretative abilities of the models and the performance of applied techniques.

5. Evaluate and demonstrate the effectiveness of the proposed models and techniques on well-known and relevant benchmarks.

1.4 Main Contributions

Diverse methodologies and techniques have been emerged throughout the course of our research. We highlight our main contributions as follows:

1. Develop the SlowFast-RULSTM, a multi-time-scale learning technique that benefits from slow and fast time branches to augment the performance of the state-of-the-art RULSTM model in action anticipation (Chapter 3).

2. Introduce early pedestrian intent anticipation, extending the standard prediction protocol of the intent prediction task (Chapter 4).

3. Develop G-RULSMT that benefits from a goal module to envision future motion in multiple feature spaces, integrating it into the observed motion, and providing more accurate action anticipations (Chapter 4).

4. Develop TAMFORMER, a multimodal transformer model for early action anticipation, that benefits from our novel adaptive temporal masking technique (Chapter 5).

5. Introduce an auxiliary loss function that allows for transfer learning between late action anticipations and early anticipations, targeting our early action anticipation task (Chapter 5).

6. Develop a novel data augmentation technique that aims to increase the amount of training data in smaller urban-scenarios datasets (Chapter 5).

7. Extend the pedestrian action prediction task to a multi-action anticipation, in contrast to the binary-prediction of crossing/not crossing (Chapter 6).

8. Introduce a text modality for anticipating the pedestrian actions, allowing for increased understanding of the observed scenes and better anticipations (Chapter 6).

9. Investigate diverse techniques for extracting and creating textual descriptions of the observed scenes (Chapter 6).

10. Contribute to the development of a dual-graph road-network anticipation approach that benefits from both the physical network graph and a contextual-correlation graph for traffic flow anticipation (Chapter 7).

11. Contribute to the study of a fast diffusion model applied to traffic data imputation (Chapter 7).

12. Investigate and develop a new semi-supervised transformer-based approach for anomaly actions recognition and localization, based on estimating the impact of anomalous events on a classified video within a temporal hierarchical patching scheme (Chpater 8).

13. Perform extensive ablation experiments to select the most appropriate construction of our models.

14. Conduct multiple evaluation experiments on popular benchmarks, comparing the proposed models with the state-of-the-art works in the target domain.

1.5 List of Publications

Most of our research has been published or is currently under review in international scientific journals or conferences, in addition to the work in-preparation for submission. Here, we list our published and under-review papers:

1.5.1 Published Papers

- **SlowFast-RULSTM (Chapter 3)**
  N. Osman, G. Camporese, P. Coscia, and L. Ballan, "SlowFast Rolling-Unrolling LSTMs for action anticipation in egocentric videos," in Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV Workshop). 2021, pp. 3437-3445 [50].

- **Early Action Anicipation with G-RULSTM (Chapter 4)**
  N. Osman, E. Cancelli, G. Camporese, P. Coscia, and L. Ballan, "Early pedestrian intent prediction via features estimation," in IEEE ICIP, 2022, pp. 3446–3450. [51].

- **TAMFORMER (Chapter 5)**
  N. Osman, G. Camporese, and L. Ballan, "Tamformer: Multi-modal transformer with learned attention mask for early intent prediction," in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023, pp. 1–5. [52].

1.5.2 Under-Review Papers

- **Traffic Flow Anticipation (Chapter 7)**
  Cheng, Shaokang, Nada Osman, Qu Shiro and Lamberto Ballan. "DSP-ST: Dynamic Structural Prior Spatio-Temporal Graph Attention Networks for Traffic Speed Prediction". Submitted to IEEE Intelligent Transportation Systems Magazine.

- **Traffic Data Imputation (Chapter 7)**
  Cheng, Shaokang, Nada Osman, Qu Shiro and Lamberto Ballan. "FastSTI: A Fast Conditional Pseudo Numerical Diffusion Model for Spatio-Temporal Data Imputation". Submitted to IEEE Transactions on Intelligent Transportation Systems.

1.6 Dissertation outline

To illustrate our research comprehensively, we started in this first chapter with a brief research background, introducing the addressed tasks and problems, outlining the main objectives and contributions of our research, and listing our produced publications. **Chapter 2** digs into the existing literature, reviewing the most influential works on human action decoding and anticipation in our target domains. Subsequent chapters, beginning with Chapter 3, systematically address specific parts of our contributions: **Chapter 3** outlines the design of the SlowFast-RULSTM model, the ablation experiments related to constructing the model, and its experimental evaluation. **Chapter 4** addresses the task of early pedestrian intent prediction with an explanation of the G-RULSTM model and its ablation and evaluation. **Chapter 5** focuses on the TAMFORMER model, again with ablation and evaluation. **Chapter 6** explains the language-aided action anticipation part of our work. **Chapter 7** explains the work conducted in traffic flow anticipation and data imputation. Finally, **Chapter 8** discusses our approach in the anomaly action recognition task, with its preliminary outcomes. The concluding chapter, **Chapter 9**, summarizes the findings and suggests directions for future research.

Chapter 2

Literature Review

Chapter Abstract

This chapter navigates through the literature of action decoding and prediction tasks, from recognition to anticipation. It starts with related works on action recognition. Then, it proceeds to the anticipation task, moving to the literature of the specific traffic anticipation task. Finally, we survey the different modeling techniques applied throughout our work, including the related vision-language integration works.

2.1 Action Recognition

Recognizing an observed action is the first step for solving more complex tasks, such as early action recognition or action prediction and anticipation. Herein, we review the literature on both fully-observed action recognition and early action recognition.

2.1.1 Fully-Observed Action Recognition

Action recognition aims to predict a labeled action category assigned to an input video. Learning from videos requires capturing both spatial and temporal information, and several approaches have been proposed to solve this task. A simple modeling strategy is based on extracting spatial features from observed video frames with a 2D Convolutional Neural Network (CNN), and their aggregation at temporal level [15, 16], or with Long-Short Term Memory (LSTM) networks

[16, 17]. Another popular approach exploits 3D CNNs where spatio-temporal information is gradually fused, leading to a better video representation and more accurate results [18–21]. Another successful idea uses two-stream networks where RGB frames and optical flow features are processed, providing more detailed motion information in input videos [16, 18, 22, 23].

A branched task from action recognition is anomaly action recognition, where it aims to classify the normality/abnormality of the observed actions. Many works tackled the task, especially in semi-supervised learning approach [24–28]. The standard methodology for the task in literature employs multiple instance learning techniques [49]. More details is provided in Chapter 8.

2.1.2 Early Action Recognition

Early action recognition is a closer step on the way of action anticipation, where it aims to classify the action of unfinished video, observing only the early frames. Some works tackled the problem using the same action recognition techniques [29], which is not the best approach for solving the problem. An early attempt for early event detection was proposed in [30], where a monotonically increasing scoring function score is presented to distinguish the start and the end of the event in order to separate the action from surrounding activities, allowing for better recognition of the action from an early stage. Most early action recognition works split the action video into a set of segments, where each segment represents a visual word, and the target is to complete the visual sentence [31–34]. The most popular technique of modeling the observed actions and their temporal relations is by utilizing LSTM networks [35–38]. Another interesting idea is the hierarchical modeling of visual segments, from coarse to fine, for a better understanding of the event and sounder prediction [39]. Another one is the integration of memory for hard-to-remember actions [40].

2.2 Action Anticipation

Different from action recognition, action anticipation aims to predict future actions _did not occur or start yet_ relying only on past video frames [41]. Previous works proposed different models for activity anticipation in third-person videos (Figure 1.9) and first-person videos (Figure 1.8).

2.2.1 Action Anticipation in Third-Person Videos

Human action anticipation started in third-person videos, where the activity of the person is captured through an external viewpoint [41,53–56]. Again, LSTM is essential in modeling the temporal dimension [41, 53]. Additionally, the encoder-decoder idea is introduced for action anticipation [41]. A recent important application of third-person action anticipation is found in urban scenario predictions and anticipation, either driver action prediction [57] or pedestrian action prediction [58].

2.2.2 Action Anticipation in First-Person Videos

In first-person, the activity is captured through an egocentric camera mounted on the head of the person performing the action. Such a perspective is important in human-robot interaction applications. Many previous works address action anticipation in first-person videos [59–63]. Our SlowFast-RULSTM and G-RULSTM models (see Chapters 3 and 4) adopt the baseline presented in [42] (Figure 2.1), where the predicted action is computed at fixed anticipation times before it starts. This task is challenging since it involves learning spatial and temporal relationships among past and future frames. To this end, [42] proposes an encoder-decoder LSTM-based architecture where past information is firstly summarized and future actions are computed leveraging features extracted from past information.

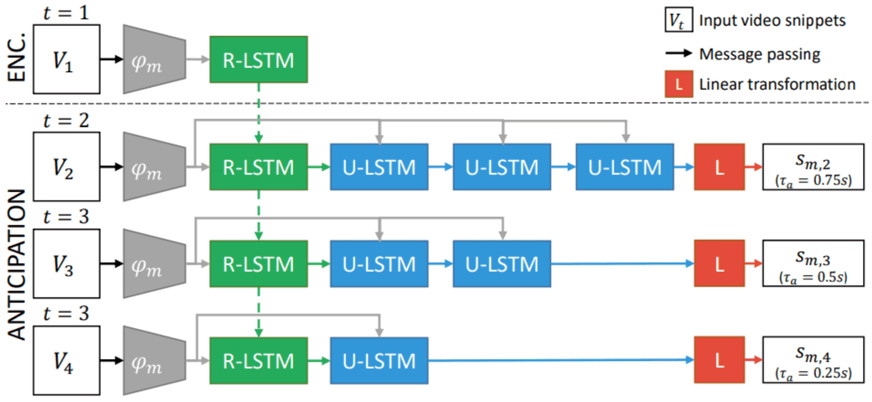


Figure 2.1: Rolling Unrolling LSTM [42]

2.2.3 Pedestrians Action Anticipation

One critical sub-problem in action anticipation is pedestrian action prediction. A self-driving car is required to predict future action, specifically, the crossing action of the pedestrians in urban scenarios, based on the observed motions of the pedestrians and the visual context of the scene. Early work in crossing prediction observed only 0.3-0.5 seconds before the crossing event to extract context features using CNN and then applying an SVM classifier to predict the crossing action in the proceeding frame [7]. More recent works extended the anticipation time to different values with different observation lengths, in addition to applying more advanced models using different types of features [64–66]. In [58], an evaluation benchmark is proposed for the crossing prediction problem, which is followed by many works, including [67–70]. This benchmark aims to predict the crossing action in a time range between 1.0 and 2.0 seconds before the event and uses an overlapped 0.5 seconds of observation. In contrast, our work aims to extend the anticipation time to 4.0 seconds, use an adaptive, not fixed, observation length, and dispense the overlapping to have an explicit performance at different anticipation times (see Chapter 4).

Another recent pedestrian motion prediction work [46] provides a broader range of pedestrian actions, such as *walking, standing, crossing, etc.*. However, the final target was not action prediction or anticipation but trajectory prediction of the observed agents. In our work, we adopt this wider range of actions for extending the action anticipation task in urban scenarios (see Chapter 6).

2.3 Traffic Flow Anticipation

Traffic flow and traffic conditions anticipation plays vital role in smart cities and intelligent transportation systems. Such importance resulting in a rich research field of interesting works.

2.3.1 Classical Approaches

Classical approaches are mainly statistical or classic machine learning methods. A set of statistical studies such as Kalman Filter [71] and Autoregressive Integrated Moving Average (ARIMA) [72] have been used to forecast traffic conditions based on the assumption of a stationary time series. In [73], an SVM model is applied for short-term traffic flow prediction, while [74] extended the SVM model to combine Bayesian classifier and SVR modeling. However, such methods require much human intervention in feature engineering; they are generally suitable for scenarios with limited road area and less training sample size.

2.3.2 Deep Learning Approaches

In recent years, deep learning-based methods have shown promising traffic prediction results due to their ability to capture and model spatial relations within the road network and temporal relations within the observation time. Convolutional neural networks (CNN) are commonly employed to extract spatial correlations in road networks [75–78]. However, the road network is a structure of discrete and irregular tomography (non-Euclidean space). Nowadays, graph convolutional networks (GCN) have drawn widespread attention because of their ability to handle arbitrary graph-structured data [48, 79–82]. This way, traffic features (e.g., traffic speed) can be propagated among graph nodes through their adjacency matrix.

For temporal modeling, recurrent neural networks (RNN) and their variants (e.g., LSTM and GRU) are typical temporal relation analysis methods for traffic prediction. A sequence of historical temporal features (e.g., traffic speed values) are fed into RNN-based models to extract the temporal correlations and produce the anticipations [48, 77, 79, 83–85].

In our work, we build upon a state-of-the-art model for traffic flow anticipation [86], a graph-based architecture with a temporal convolution network, in which we adapt our dual-graph approach (see Chapter 7).

2.4 Multi-Modal Multi-Scale Modeling

Multi-modal and Multi-scale modeling are powerful design paradigms that empower a hidden input representation to maximize the valuable information and to be more robust to scale changes with respect to a single-modality or a single-scale modeling approach. The multi-modal technique allows the beneficiation of different types of extracted information as long as the proper technique is used to fuse modalities. On the other hand, multi-scale modeling is another approach for maximizing the understanding of the hidden input, and it can be adopted in both spatial and temporal domains. In our work, we employ both of the modeling techniques.

2.4.1 Multi-Modal Modeling

Multi-modal modeling is becoming very popular due to its effectiveness in enriching the exacted information by merging multiple views about the observation.

Each modality catches distinctive details on the observed input. Often, different modalities are complementary to each other, where fusing the multi-modal feature spaces would comprehensively describe the observed scene [87,88]. Many previous works adopted a multi-modal design and proved an improved performance in the action anticipation and detection task [42, 58, 89]. Many techniques have been proposed for the fusion of multiple modalities; for example, in [42], an attention-based approach is used for fusing the different modalities. Another interesting and popular technique targeting transformer-based models is proposed in [90]. In our work, we adopt the attention-based fusion strategy [42] in the SlowFast-RULSTM and the G-RULSTM model (Chapter 3 and Chapter 4 respectively), while adopting the transformer-based fusion technique [90] in our TAMFORMER model (Chapter 5 and Chapter 6).

2.4.2 Spatial Scale Modeling

A well-known concept in computer vision is how spatial scale affects the information extracted from an image. Larger scales allow for more details but could incorporate more noise. On the other hand, smaller spatial scales could miss the key-hidden input in the image. Consequently, many works adopted multi-spatial-scale design to create a more robust model to scale changes, especially in image classification and object detection problems [91–96]. In the context of our human action anticipation task, the spatial scale controls the amount of information extracted from the surrounding environment, affecting how the model interprets the scene and predicts the future. Therefore, works addressing action prediction, mainly in urban environments, studied the effect of the spatial scale on the prediction performance [8, 66]. In our work, we consider multi-spatial-scale in Chapter 6 to allow the generation of more robust textual descriptions of the images.

2.4.3 Temporal Scale Modeling

Like spatial scale, temporal scale represents another aspect of interpreting and understanding videos. On the time dimension, not all the actions have the same progressing scale. Some activities could show fast changes; other activities could be slower. Consequently, the temporal scale affects the robustness of the model to different actions. Based on that, recent recognition and anticipation models adopt multi-time-scale modeling [97,98]. Slow-Fast networks [97] for video recognition builds benefit from processing video sequences at slow and fast frame rates with two separate branches that capture patterns at different time resolutions. In our SlowFast-RULSTM (Chapter 3), we take advantage of this idea, aiming at capturing slow and fast features for anticipating future actions. Additionally, in our TAMFORMER model (Chapter 5), we address the multi-time-scale modeling with our temporal adaptive masking technique.

2.5 Language in Vision Tasks

Language provides a rich and contextual way to describe visual content, where it can provide additional context and details that may not be immediately apparent by visual feature extraction. Therefore, coupling language features and visual features has gained massive attention in computer vision, where Numerous studies have proposed innovative techniques for such coupling and for extracting suitable language features for visual tasks. For example, in [99], a study is conducted to measure the effectiveness of different language models and text embedding approaches on different visual tasks. While in [100], VilBERT presents a co-attention transformer model that couples text and images into a multi-modal transformer to solve different vision tasks, including visual question answering and caption-based image retrieval.

The significant interest in this field has led to the production of breakthrough models, such as CLIP [101], a transformer-based model aiming at coupling textual inputs with images for zero-shot text/image retrieval. In addition to the impressive advances in natural language processing (NLP), with the witnessed improvements of the GPT models [102], and the image captioning models, such as BLIP [103,104], that applies a two-stage of training: representation learning stage of the image, and a generative stage of the textual caption to the image.

Given such advances in the language-visual models, visual tasks have started integrating language and textual input to enhance their performance. Specifically, for the task of action anticipation, recent models proposed the generation of textual labels and their integration in the anticipation model [105–109]. VLMAH [105] anticipates first-person actions, integrating images and textual descriptions of the observed actions and applying bidirectional LSTM encoders for visual and textual inputs. While in [106], a large language model is utilized to predict future actions based on extracted textual descriptions of the input images, where the visual task is fully transformed into a lingual task. A similar idea is applied in [107], proposing knowledge distillation from the lingual anticipation model to the visual anticipation model. Again, a large language model is used for anticipation in [108] using image captions generated for the input images. Another interesting technique is proposed in [109], where CLIP encodes images and textual labels of the observed actions, followed by a transformer aggregation anticipation model.

Currently, the integration of language into the action anticipation task targets mainly the anticipation of first-person actions, where the observed contexts incur many similarities between the taken actions, and language descriptions could provide more into-the-point representation in this case. In our work, we aim to integrate language into the third-person pedestrian action anticipation task, where the decisions of pedestrians are affected by many factors, such as the current action of the person, her/his demographic state, the collective behavior of surrounding people, and the state of the overall scene. Therefore, the accurate application of language can better catch such various factors compared to the sole visually extracted representations. However, this direction faces the challenge of providing the textual ground truth, especially in the complex environment of urban scenes, where we address this in our work in Chapter 6.

Chapter 3

SlowFast Rolling-Unrolling LSTMs for Action Anticipation

Chapter Abstract

This chapter explains the design of our SlowFast-RULSTM model, as we propose a novel attention-based technique to simultaneously evaluate the slow and fast features extracted from three different modalities, namely RGB, optical flow, and extracted objects. Two branches process information at different time scales, and several fusion schemes are considered to improve the anticipation accuracy. Extensive experiments are provided on in-kitchen activities datasets, EPIC-KITCHENS-55 and EGTEA Gaze+, to demonstrate the effectiveness of our technique. In addition to a detailed ablation study. The work in this chapter has been published in [50]. [*]

3.1 Introduction

Human action anticipation in egocentric videos [110–112] is a popular computer-vision research topic due to a wide range of involved domains, where we focus in this chapter on the kitchen environment anticipation. In this context, egocentric videos have provided a considerable amount of information to be used for training

[*] This chapter has been published as "SlowFast Rolling-Unrolling LSTMs for Action Anticipation in Egocentric Videos" in *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV Workshop), 2021*

action anticipation models thanks to low-cost wearable devices that offer different streams to be used [113, 114], *e.g.*, RGB videos, audio or depth data.

State-of-the-art action anticipation approaches [98, 115] are mainly based on attention mechanisms to efficiently extract relationships across the frames at a specific frame rate. Nevertheless, action speed may differ based on the actor, surrounding environment, and the action itself. To anticipate future actions, two main factors should be taken into account: window size (*i.e.*, the number of current and past actions to be considered) and processing frame rate (*i.e.*, the quantity of information to be extracted from each action). While the former is typically fixed for a fair results comparison, the latter can be arbitrarily selected. In this case, a different choice of this parameter may lead to completely different results. We demonstrate that if multiple streams of the same modality are provided to an action anticipation model, it is able to appropriately select which stream to focus on and improve its predictive capabilities, leading to a better generalization.

Based on this idea, we propose to consider multiple branches for each input modality, which process the corresponding stream at different frame rates. Herein, we focus on two popular egocentric datasets, EPIC-Kitchens-55 and EGTEA GAZE+. Based on RU-LSTM [42] model, we propose a slow-fast architecture that learns from input videos at two different scales, as shown in Figure 3.1. A slow branch processes input videos with a low frame rate, while another branch uses a higher frame rate. In this way, redundant information is discarded for actions that evolve slowly, while retained for faster actions. In order to combine these two branches efficiently, we use an attentive-based mechanism that efficiently weights their output scores and provides only one result, which is subsequently decoded

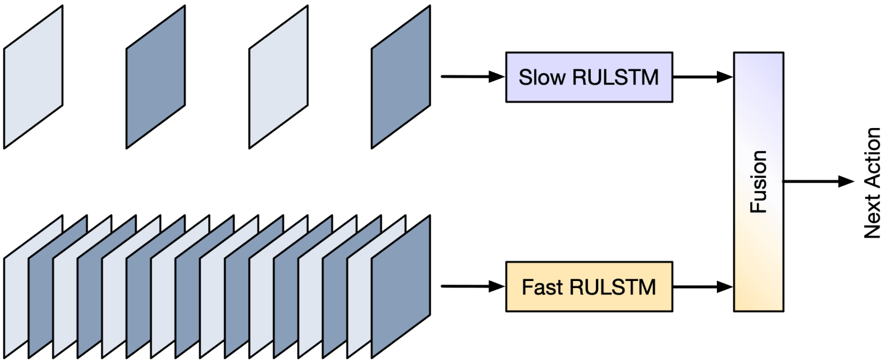


FIGURE 3.1: Human actions happen at different speeds, requiring a multi-scale approach for better predicting future behaviors. We propose a Slow-Fast RU-LSTM model containing two branches, namely *slow* and *fast* branches, which learn independently from input video features at different time scales.

to extract future actions. We show that our model systematically outperforms state-of-the-art models at different anticipation times.

The main contributions of this chapter can be summarized as follows:

1. We propose a multi-time-scale learning technique that benefits from a slow and fast branch to augment the performance of the RU-LSTM model;
2. We perform extensive ablation experiments in order to select the most appropriate frame rates and window sizes;
3. We conduct multiple evaluation experiments on popular action anticipation benchmarks and also compare different model architectures and slow-fast fusion mechanisms.

3.2 Rolling-Unrolling LSTM

Our technique is built upon the RU-LSTM [42] model, which processes sequences of feature vectors computed from input video frames. This model defines an encoding stage of S_{enc} steps and an anticipation stage of S_{ant} steps for a total of $\alpha \cdot (S_{enc} + S_{ant})$ seconds, where α is the time interval between two subsequent frames. This model is based on an LSTM-based encoder, named *rolling* LSTM (R-LSTM), and an LSTM-based decoder, called *unrolling* LSTM (U-LSTM). The former summarizes, during the encoding and anticipation stages, past information extracted from input videos and provides the latter a valuable context for predicting future action. In the anticipation stage, the decoder receives the representation from the encoder and, using the last observation computes a plausible distribution over future action classes. The encoding-decoding process is performed for each time step in the anticipation stage, and the network is trained to predict the actual action label using a cross-entropy loss. RU-LSTM processes multi-modal features combined using a mixture-of-experts-based method named Modality Attention (MATT) to exploit more context and create a more informative hidden representation. Since this model shows remarkable performance in predicting future actions from multi-modal input streams, we extend its predictive capability by explicitly designing a multi-scale fusion mechanism to capture slow and fast features from observed video sequences.

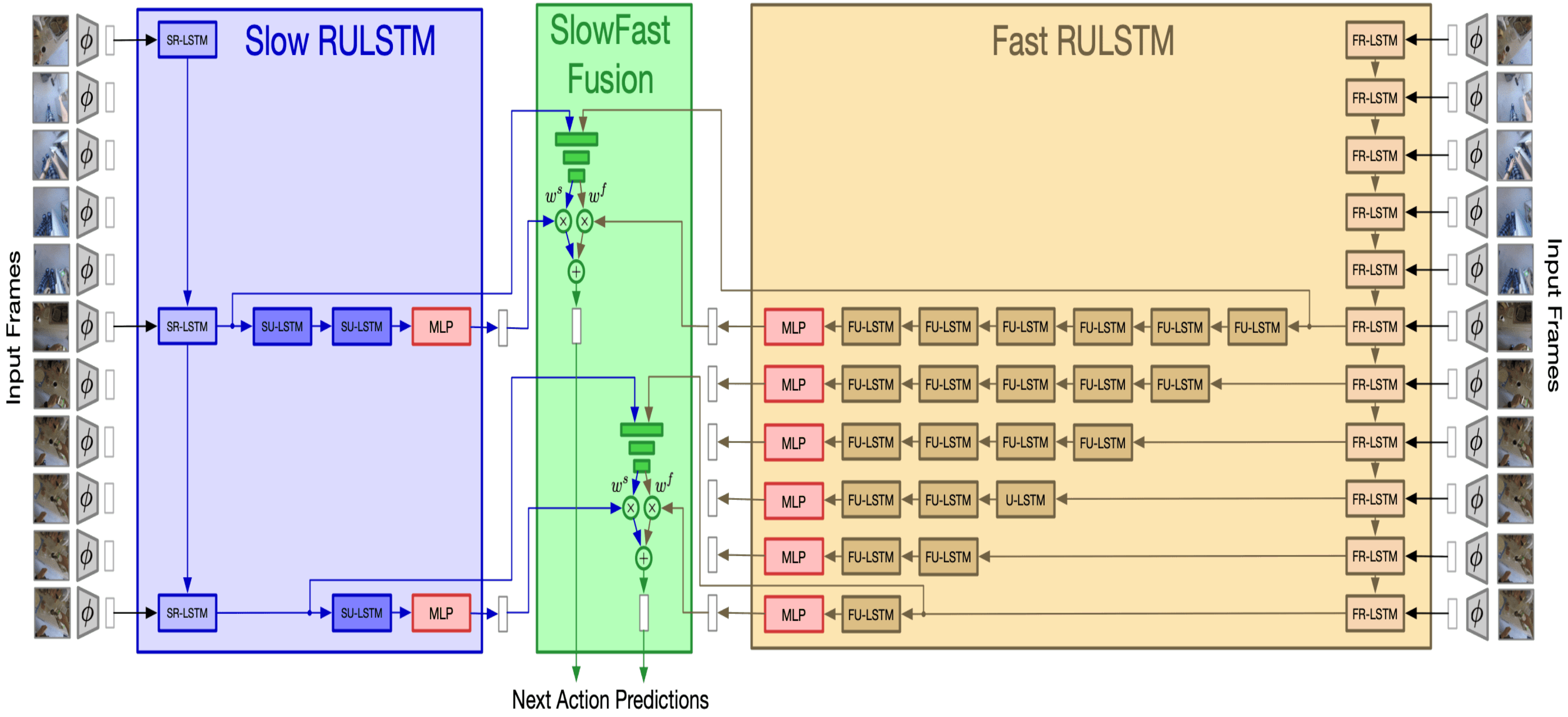


Figure 3.2: Our SlowFast RULSTM model. A CNN feature extractor first processes input videos, and then sequence representations are fed to two branches processing information at two different frame rates. Our slow and fast branches are based on RU-LSTM architecture that encodes past information and then decodes future actions. To better capture the correlations in past observed frames, we design a slow-fast fusion mechanism that merges the predictions of these two branches, leading to better accuracy.

3.3 SlowFast RULSTM

As depicted in Figure 3.2, our SlowFast RULSTM model consists of two branches: a slow branch that processes input videos using a low frame rate (one frame every α_s seconds) and a fast branch, which uses a high frame rate (one frame every α_f seconds). Our idea is to process input features at different time resolutions in order to capture slow and fast relations between past and future frames.

Let $\boldsymbol{x} \in \mathbb{R}^{T \times C \times H \times W}$ be the input video to be processed and $\boldsymbol{z} \in \mathbb{R}^{T \times D}$ the corresponding representation computed at each time step. Given a single input frame $\boldsymbol{x}_t$ at time t, $\boldsymbol{z}_t = \phi(\boldsymbol{x}_t)$ is its related representation where ϕ is a CNN feature extractor, and $T = S_{enc}+S_{ant}$ the total sequence length. Our slow branch processes input video frames at $1/\alpha_s$ frame rate while our fast branch at $1/\alpha_f = R/\alpha_s$ with $R = \alpha_s/\alpha_f$ being the ratio between fast and slow frame rates, respectively. Given an internal representation $\boldsymbol{z}_t$, the encoder in the fast branch produces feature representations used by the decoder as follows:

$$\boldsymbol{r}_t^f = \textit{FR-LSTM}\left(\boldsymbol{z}_t, \boldsymbol{r}_{t-1}^f\right) \tag{3.1}$$

where $t \in \{1, 2, \ldots, T\}$ and $\boldsymbol{r}_t^f = (\boldsymbol{h}_t^f, \boldsymbol{c}_t^f)$ is the state that contains hidden and context vectors of FR-LSTM with $\boldsymbol{h}_t^f, \boldsymbol{c}_t^f \in \mathbb{R}^d$. Our slow branch is similarly defined:

$$\boldsymbol{r}_t^s = \textit{SR-LSTM}\left(\boldsymbol{z}_t, \boldsymbol{r}_{t-1}^s\right) \tag{3.2}$$

Where $t = kR + 1$ with $k \in \{0, 1, \ldots, \lfloor T/R \rfloor\}$, and $\boldsymbol{r}_t^s = (\boldsymbol{h}_t^s, \boldsymbol{c}_t^s)$ is the state containing hidden and context vectors of slow R-LSTM with $\boldsymbol{h}_t^s, \boldsymbol{c}_t^s \in \mathbb{R}^d$.

The decoder in the fast branch receives the representations given by the fast encoder and produces the prediction by unrolling the Fast U-LSTM for $T - t + 1$ steps as follows:

$$\boldsymbol{u}_{t,q}^f = \textit{FU-LSTM}\left(\boldsymbol{z}_t, \boldsymbol{u}_{t,q-1}^f\right), \tag{3.3}$$

$$\boldsymbol{u}_{t,t-1}^f = \boldsymbol{r}_t^f, \qquad \boldsymbol{u}_t^f = \boldsymbol{u}_{t,T}^f, \tag{3.4}$$

where $q \in \{t, \ldots, T\}$. Then, a fast prediction score over all action classes is computed from the output of the decoder with a Multi-Layer Perceptron (MLP) at each time step as $\boldsymbol{l}_t^f = MLP(\boldsymbol{u}_t^f)$, where hidden and context vectors in $\boldsymbol{u}_t^f$ are concatenated. Similarly, the slow decoder receives the slow encoded features $\boldsymbol{r}_t^s$ and produces $\boldsymbol{u}_t^s$ by unrolling the Slow U-LSTM and then slow logits $\boldsymbol{l}_t^s$ are computed with an MLP. The formulation related to the slow decoding step is as follows:

$$\boldsymbol{u}_{t,q}^s = SU\text{-}LSTM\left(\boldsymbol{z}_t, \boldsymbol{u}_{t,q-1}^s\right), \tag{3.5}$$

$$\boldsymbol{u}_{t,t-1}^s = \boldsymbol{r}_t^s, \qquad \boldsymbol{u}_t^s = \boldsymbol{u}_{t,T}^s, \tag{3.6}$$

$$\boldsymbol{l}_t^s = MLP\left(\boldsymbol{u}_t^s\right). \tag{3.7}$$

After slow and fast logit scores computation, our model fuses the obtained predictions with an attention mechanism. Specifically, given both slow and fast scores ($\boldsymbol{l}_t^s$ and $\boldsymbol{l}_t^f$), we compute our final merged logits as $\boldsymbol{l}_t = w_t^s \cdot \boldsymbol{l}_t^s + w_t^f \cdot \boldsymbol{l}_t^f$, where w_s and w_f represents slow and fast multipliers that weight slow and fast predictions computed as follows:

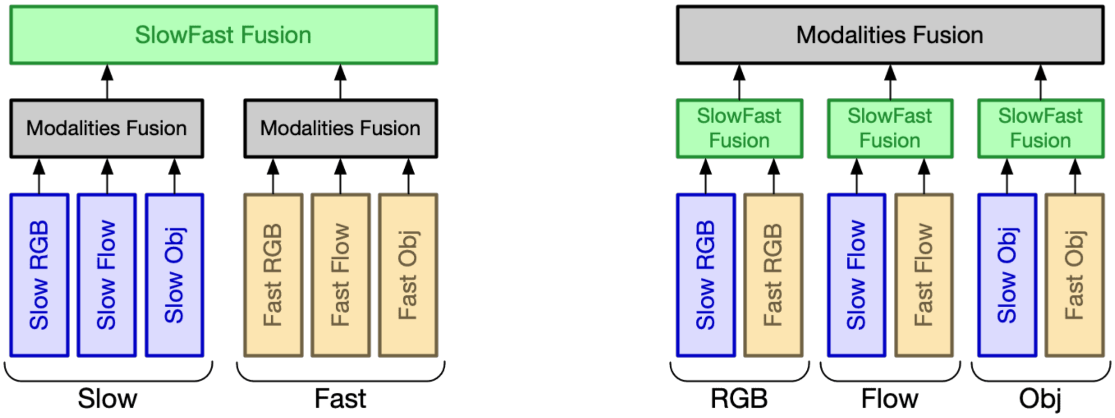
$$\left[\lambda_t^s, \lambda_t^f\right] = \text{MLP}\left(\left[\boldsymbol{r}_t^s, \boldsymbol{r}_t^f\right]\right), \tag{3.8}$$

$$w_t^s = \frac{e^{\lambda_t^s}}{e^{\lambda_t^s} + e^{\lambda_t^f}}, \qquad w_t^f = \frac{e^{\lambda_t^f}}{e^{\lambda_t^s} + e^{\lambda_t^f}}, \tag{3.9}$$

where $\left[\ \cdot\ \right]$ stands for the concatenation operator.

3.4 SlowFast and Modalities Fusion Strategies

As proposed in [42], anticipating future actions can take advantage of multi-modal input representations. For this reason, RU-LSTM proposes an attention mechanism (MATT module) that appropriately weights each input modality. In our work, we exploit the multi-modal video representation and investigate two different techniques to embed both multi-modal and multi-scale inputs. As shown in Figure 3.3, we could either merge our modalities with a MATT module and then fuse both slow and fast branches (Fig. 3.3a) or firstly fuse slow and fast branches for each modality, and then merge with a MATT module the multi-modal representations (Fig. 3.3b). More specifically, Figure 3.3a depicts an architecture that



(A) Mod-SF Fusion

(B) SF-Mod Fusion

FIGURE 3.3: SlowFast and Modalities fusion schemes. (a) Modalities fusion is applied at slow and fast frame rates, and then SlowFast fusion is applied to the fused modalities. (b) SlowFast fusion is first applied to each modality separately, and then the modalities fusion is applied to the fused time scales.

fuses two RU-LSTMs trained on two different time scales with our slow-fast attention scheme. The input of the attention network is the concatenation of the time scale branches, where each branch is represented by the weighted internal representation $\boldsymbol{r}_t$ of the R-LSTM encoders for all the modalities, using the pre-trained modalities attention weights.

As discussed in Sec 3.3, in Figure 3.3b, each modality is trained with a slow and fast branch, fused with the slow and fast module, and then each modality is merged with the same MATT used in RU-LSTM.

3.5 Experimental Results

We conducted several experiments on two popular datasets used for action anticipation in order to investigate our SlowFast RULSTM model. Furthermore, we study two architectures that embed different fusion mechanisms dealing with multi-modal and multi-scale inputs. In the following, we describe our datasets, the evaluation metrics, and the experiments in order to show the impact of our slow and fast modeling approach.

3.5.1 Implementation Details

We use PyTorch [116] for our implementation and use pre-extracted features provided by [42] for training our method. We found it beneficial to train each branch separately and then fine-tune at the fusion stages. Specifically, for the Mod-SF Fusion approach (see 3.3a), we train RU-LSTM at different frame rates using its standard training pipeline, applying sequence completion pre-training stage and then fine-tune slow and fast branches at the final stage. For the SF-Mod Fusion approach, we apply a similar training strategy.

3.5.2 Datasets

As described in Subsection 1.2.1, we experiment on two popular egocentric datasets: EpicKitchens-55 [43] and EGTEA Gaze+ [44]. EpicKitchens-55 collects 55 hours of recorded videos and $39,596$ annotations of 32 participants involved in their daily kitchen activities. The annotations contain 125 verb and 352 noun classes. All unique (*verb*, *noun*) pairs are considered for a total of $2,513$ unique action labels. EGTEA Gaze+ contains 28 hours of video clips showing hand-object interaction actions performed by 32 participants. It contains 19 verbs, 51 nouns, and 106 unique actions. The average across three splits reported by the authors of the dataset is considered.

We evaluate our proposed SlowFast RULSTM model for both datasets using a Top-5 accuracy metric at different anticipation times.

3.5.3 Quantitative Results

3.5.3.1 Evaluation Results on EpicKitchens-55

Table 3.1 reports our results for SlowFast RULSTM and RU-LSTM models on the validation split of the EpicKitchens-55 dataset. Our method outperforms RU-LSTM considering both each modality separately and their fusion. The RGB branch shows an improvement of 1.22% at 1 s. Additionally, almost 1% of improvement is achieved for both FLOW and OBJ modalities. Combining all modalities, our model achieves a 36.09% anticipation accuracy at 1 s, with an improvement of approximately 0.8% over the RU-LSTM baseline. Our model also shows a remarkable gain at 2 s of 1.14%, validating our idea to use a multi-scale approach for capturing more information at the early stages of action anticipation. Our results prove that processing egocentric videos at different frame rates improves the prediction accuracy, as the model benefits from the different information extracted at different time scales.

Top-5 ACTION Accuracy% @ different τ_a(s)					
		2.0	1.5	1.0	0.5
RGB	RULSTM [42]	25.44	28.32	30.83	33.31
	SF-RULSTM	**26.78**	**29.25**	**32.05**	**34.34**
	Imp.	+1.34	+0.93	+1.22	+1.03
FLOW	RULSTM [42]	17.38	18.91	21.42	23.49
	SF-RULSTM	**18.01**	**19.82**	**22.36**	**24.15**
	Imp.	+0.63	+0.91	+0.94	+0.66
OBJ	RULSTM [42]	24.56	26.61	29.89	31.82
	SF-RULSTM	**25.61**	**27.64**	**30.8**	**32.15**
	Imp.	+1.05	+1.03	+0.91	+0.33
FUSION	RULSTM [42]	29.44	32.24	35.32	37.37
	SF-RULSTM	**30.58**	**32.83**	**36.09**	**37.87**
	Imp.	+1.14	+0.59	+0.77	+0.5

TABLE 3.1: Top-5 accuracy at different anticipation times for RU-LSTM and our SF-RULSTM model.

Top-5 ACTION Accuracy% @ 1s				
	RGB	FLOW	OBJ	FUSION
TAB	28.25	19.60	30.09	35.73
SF-RULSTM	**32.05**	**22.36**	**30.8**	**36.09**
Imp.	+3.8	+2.76	+0.71	+0.36

TABLE 3.2: Comparison of action anticipation Top-5 accuracy at 1 s between SF-RULSTM and TAB [98] model.

Table 3.2 reports a comparison between SlowFast RULSTM and Temporal Aggregation Block (TAB) models, as proposed in [98], which is a current state-of-the-art multi-scale approach for action anticipation. We report results at anticipation accuracy of 1 s, as authors do not provide anticipation accuracy at different anticipation times. TAB performance is obtained by using the same configuration reported in [98]. Our results show an accuracy improvement for RGB and FLOW modalities of +3.8% and +2.76%, respectively. In this case, our improvement for both the OBJ modality and the complete model is less marked, yet our slow-fast fusion model still outperforms the TAB model.

Top-5 ACTION Accuracy% @ different τ_a(s)					
		2.0	1.5	1.0	0.5
RGB	RULSTM	56.41	60.68	66.76	72.04
	SF-RULSTM	**57.84**	**62.36**	**67.21**	**72.32**
	Imp.	+1.43	+1.68	+0.45	+0.28
FLOW	RULSTM	33.92	35.83	39.51	42.62
	SF-RULSTM	**36.93**	**39.29**	**42.84**	**45.94**
	Imp.	+3.01	+3.46	+3.33	+3.32
FUSION	RULSTM [42]	56.82	**61.42**	66.4	71.84
	SF-RULSTM	**57.48**	61.37	**67.6**	**72.22**
	Imp.	+0.66	-0.05	+1.2	+0.38

Table 3.3: Top-5 accuracy at different anticipation times for EGTEA Gaze+ dataset.

3.5.3.2 Evaluation Results on EGTEA Gaze+

Table 3.3 compares our proposed SlowFast RULSTM model to the RU-LSTM model on the EGTEA Gaze+ dataset. Only RGB and optical flow features are available for this dataset, and we train the RU-LSTM model to obtain results for both modalities. By contrast, RU-LSTM fusion results are reported from [42]. The table shows a maximum improvement for the FLOW modality of approximately +3.5% at 1 *s*. Furthermore, our complete model improves the anticipation accuracy at 1 *s* by 1.2%, which can be considered a relevant gain due to the reduced number of classes of this dataset compared to EpicKitchens-55 (106 instead of 2513 classes).

3.5.4 Ablation Experiments

To assess the performance of each part of our model, we conduct a set of ablative experiments. In this case, we focus on the EpicKitchens-55 dataset. Additionally, all single modality-related experiments use only RGB features, as they can be assumed to be more inclusive features than both optical flow and object-based features.

3.5.4.1 Selection of Time Step Value

The main element of our model is represented by the choice of slow and fast time steps to be fused. Table 3.4 illustrates our anticipation accuracy using different

	Top-5 ACTION Accuracy% @ different τ_a(s)							
1 α (s)	2.0	1.75	1.5	1.25	1.0	0.75	0.5	0.25
0.1	25.13	-	27.26	-	30.44	-	33.27	-
0.125	24.53	25.63	27.3	**28.97**	**30.96**	**32.23**	**33.49**	**35.02**
0.2	25.16		-	-	30.71	-	-	-
0.25	25.2	**25.84**	27.78	28.84	30.55	31.92	33.19	34.43
0.5	**26.39**	-	**28.4**	-	**30.94**	-	32.87	-
1.0	25.56	-	-	-	30.13	-	-	-

TABLE 3.4: Top-5 accuracy at different time steps (α) for a single modality (RGB). At 1 s the best performance is achieved considering two frame rates: 0.125 and 0.5.

time steps ($\alpha \in \{0.1, 0.2, 0.25, 0.5, 1.0\}$) for RGB features. As shown, the best results (at 1 s) are obtained selecting $\alpha = 0.125\ s$ and $\alpha = 0.5\ s$. For this reason, we use these two values for our fast and slow branches, respectively.

Additionally, Figure 3.4 compares Top-5 accuracy results for each modality, using three different time steps: 0.125 and 0.5, as obtained by our previous experiments for the RGB modality, and 0.25, which represents the default time step value used in [19]. As shown, our selected time steps improve Top-5 accuracy for each modality.

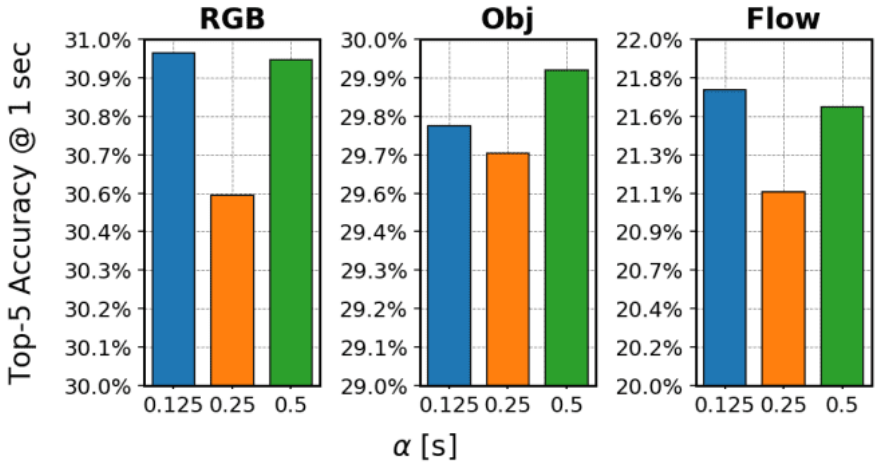


FIGURE 3.4: Top-5 accuracy varying the time step α for different input modalities. We select $\alpha \in \{0.125, 0.5\}$ for our SlowFast architecture as each branch appears more accurate with respect to selecting $\alpha = 0.25$, as used in [42].

Top-5 ACTION Accuracy% @ 1s		
τ_e (s)	$\alpha = 0.125$	$\alpha = 0.5$
1.5	**30.96**	30.94
3.0	30.66	**31.44**

Table 3.5: Action anticipation results at 1 s for two different lengths of encoding time (τ_e) for RGB modality.

3.5.4.2 Sequence Length Encoding

Extracting relevant features from a video sequence may not only depend on the selected frame rate but also on the length of observed sequences. To this end, we test the impact of different buffer lengths on the anticipation task for the RGB features. Two buffer lengths are considered: $\tau_e = 1.5\ s$, as proposed in [42], and $\tau_e = 3.0\ s$. As shown in Table 3.5, increasing the buffer length provides a noticeable improvement for the slow model ($\alpha = 0.5\ s$), while the opposite arises for the fast model ($\alpha = 0.125\ s$). Since the slow model processes a smaller number of video frames, it seems to be able to store more past frames. By contrast, increasing the buffer of the fast model increases its complexity, requiring a smaller window size to achieve better results.

3.5.4.3 SlowFast Fusion

Table 3.6 reports our results for different slow-fast fusion schemes considering the RGB modality. The first three rows show different fusion methodologies using two scale branches: slow (with $\alpha = 0.5s$) and fast (with $\alpha = 0.125s$). We consider two additional fusion techniques other than the proposed attention-based fusion:

- *Concat*: prediction obtained directly from the concatenation of the internal representations of the slow and fast branches;
- *Ensemble*: average of the predictions of the slow and fast branches.

The best fusion scheme at 1 s is represented by an attention-based approach, which appears to discriminate better which branch should be used more for predicting future actions. The last row reports our results for the attention-based model considering an additional time scale branch ($\alpha = 0.25\ s$). These results confirm that anticipating future human actions requires different time scales for obtaining better performance. Among the proposed models, the best results are

Top-5 ACTION Accuracy% @ different τ_a(s)					
	α(s)	2.0	1.5	1.0	0.5
Concat	{0.125, 0.5}	24.59	26.9	30.04	32.73
Ensemble (AVG)	{0.125, 0.5}	26.98	29.59	31.71	34.2
Attention	{0.125, 0.5}	26.78	29.25	**32.05**	34.34
Attention	{0.125, 0.25, 0.5}	26.84	29.51	31.91	33.96

Table 3.6: Top-5 accuracy at different anticipation times for different slow-fast fusion schemes (RGB modality).

Top-5 ACTION Accuracy% @ 1s	
Concatenation	31.92
Mod-SF Fusion	**36.09**
SF-Mod Fusion	35.28

Table 3.7: Top-5 ACTION accuracy at 1 s for different variations of modalities fusion.

achieved using two scale branches (slow and fast), while adding another branch does not provide any improvement.

3.5.4.4 Modalities Fusion

To assess the performance of the proposed modalities fusion mechanism, shown in Figure 3.3a (Mod-SF Fusion), an alternative fusion architecture (SF-Mod Fusion) is tested (see Figure 3.3b). Table 3.7 provides Top-5 accuracy for both Mod-SF Fusion and SF-Mod Fusion approaches. Additionally, we change the slow-fast attention input to the concatenation of the internal representations of all R-LSTM branches instead of the weighting mechanism, as discussed in 3.4. As shown, the Mod-SF Fusion approach appears to be the best configuration since it is easier, compared to the other models, to combine different modalities and allow them to aid each other. Using SF-Mod Fusion, the combination of multi-modal predictions is more complex and reduces model performance. The approach based on concatenation provides the lowest accuracy, which can be due to the huge input size of the attention network.

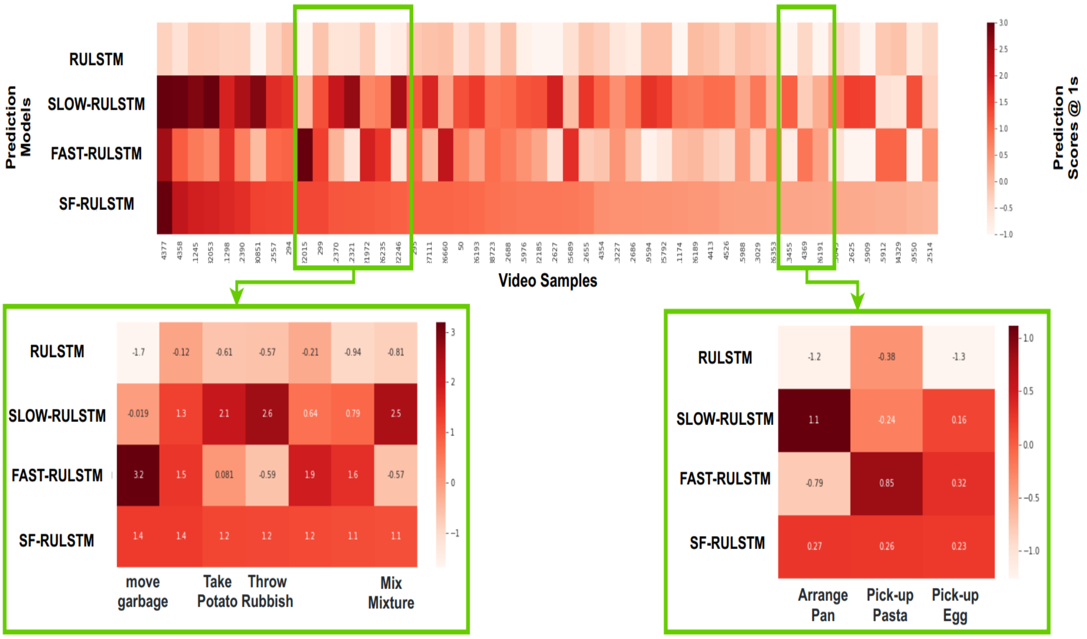


Figure 3.5: Predictions scores of different video samples from our validation set, where our model provides higher prediction scores than the RU-LSTM model. For many actions (*e.g.*, *move garbage, arrange pan*), at least one slow/-fast branch has a higher prediction score, and so is our complete slow-fast model compared to the selected baseline.

3.5.5 Qualitative Results

We qualitatively evaluate the behavior of our proposed SF-RULSTM in Figure 3.5 and Figure 3.6. Figure 3.5 shows the prediction scores of our SF-RULSTM model (last row) against RU-LSTM model scores (first row) considering a subset of validation samples, *i.e.*, the ones where RU-LSTM assigns low scores. By contrast, our model benefits from either the slow branch (second row) or the fast branch (third row), resulting in a higher score.

Finally, Figure 3.6 shows how the slow-fast attention model adapts to different action speeds. Our model is able to select the most appropriate branch for the current action speed, *i.e.*, the slow one when limited changes in the RGB video stream occur or the fast branch for actions that evolve more rapidly.

3.6 Conclusion

This chapter proposes a multi-time-scale attention-based approach to fuse information extracted at different time scales for anticipating human actions in egocentric videos. Two branches process input videos to capture slow and fast features and

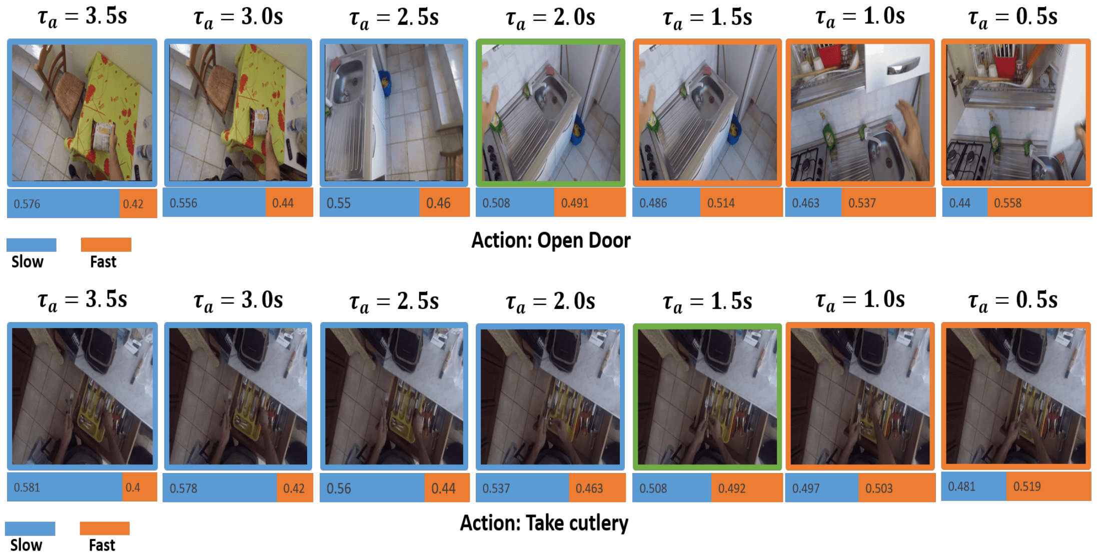


FIGURE 3.6: Two examples of actions where slow-fast attention weights change over time. The actions start with no significant changes in the input frames, so the attention mechanism weighs more in the slow branch. When the action rapidly evolves, more attention is provided to the fast branch instead.

better discriminate among different actions (or the same action performed by different actors). We design several fusion techniques for combining multiple input modalities and demonstrate that an anticipation model can benefit from fusing input modalities before combining different time scales.

We outperform a state-of-the-art model on two popular kitchen activities benchmarks, *e.g.*, EpicKitchens-55 and EGTEA GAZE+, and show better results compared to a multi-scale model on the EpicKitchens-55 dataset. Our future work will focus on considering more branches and investigating new techniques to combine several multi-scale branches better.

Chapter 4

Early Pedestrian Intent Prediction Via Features Estimation

Chapter Abstract

This chapter addresses our second action anticipation domain of pedestrian action anticipation, explaining the G-RULSTM model. Our work primarily aims to revise this standard evaluation protocol to forecast crossing events as early as possible, where we conceive a solution upon RULSTM, proposing to envision future features or modalities through a goal module to better infer human intentions, using a properly attention-based fusion mechanism. Experimental results validate our model against JAAD and PIE datasets and demonstrate that an intent prediction model can benefit from these additional clues for anticipating pedestrian crossing actions. The work in this chapter has been published in [51]. [*]

4.1 Introduction

Self-driving cars and autonomous robots have recently been demonstrated to be capable of performing multiple tasks (*e.g.*, planning, control, manipulation, and

[*] This chapter has been published as "Early Pedestrian Intent Prediction" in *Proc. of IEEE ICIP International Conference on Image Processing, 2022.*

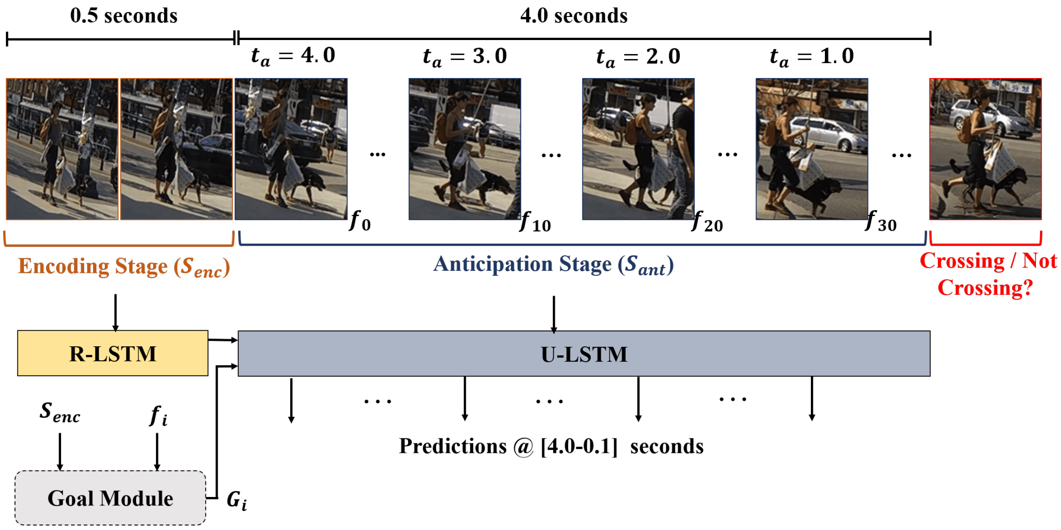


FIGURE 4.1: Our model detects crossing events in two stages: an encoding stage (R-LSTM), processing the initial part of the sequence (0.5 s), and a decoding stage (U-LSTM), predicting the event at multiple anticipation times. We estimate future visual and non-visual features with an attention-based (goal) module that is provided for the decoding stage. We consider anticipation times in the range $[4.0 - 0.1]$ seconds.

perception). Nevertheless, in crowded scenarios, such as streets, parks, or airports, they must face critical decisions to avoid accidents and perform smooth navigation. For example, assistive or delivery robots need to anticipate human motion to plan their motion better and be socially compliant. In this regard, urban contexts represent relevant scenarios where predicting human intentions relies on both fast and correct scene perception. Furthermore, predicting future intentions as early as possible helps in better plan their interaction with the environment. Multiple cues are typically involved in this process, *e.g.*, relative speed, pedestrian pose, and road signs [7, 64–66].

In this regard, we primarily focus on predicting pedestrians' intentions to cross-roads as early as possible, given a fixed history of frames. To this end, we build on top of an action anticipation model, an architecture that takes multiple input features and provides the probability that each monitored pedestrian will either be crossing or not in the future. Using past motion along with the surrounding context is extremely useful when a pedestrian is walking on a sidewalk prior to crossing or standing due to low visibility on rainy or foggy days, for example. To improve our prediction accuracy, we conceive a *goal* module, whose aim is to predict future features to be fused to motion history (see Fig. 4.1).

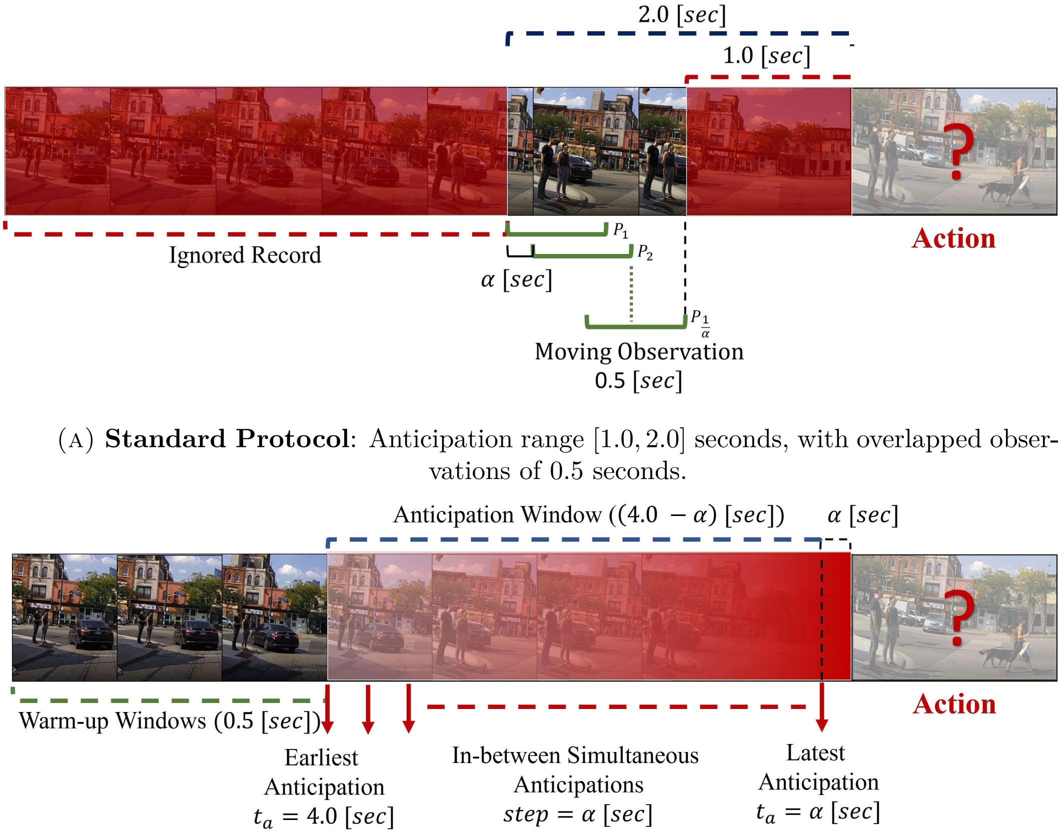
Predicting if a pedestrian will cross or not is still a challenging problem. Prior works on intent crossing prediction only focus on a time window of 0.3-0.5 s before the event to extract context features and classify the event [7]. More recent works extend this anticipation time to different values with different observation lengths, in addition to developing more advanced models with multiple input features [64–66]. In [58], an evaluation benchmark is proposed to tackle this problem, also adopted in [67–70]. This benchmark focuses on predicting future intentions between 1.0 to 2.0 s earlier than the event and uses overlapping windows of 0.5 s as motion history. An averaging operation is then employed to obtain the final prediction. By contrast, our work aims to extend this anticipation time from 1.0 s to 4.0 s and use an adaptive observation window. We do not employ an averaging operator to extract predictions at fixed anticipation times yet use pedestrian records as single samples. We mainly focus on two largely adopted datasets, namely Joint Attention for Autonomous Driving (JAAD) [7] and Pedestrian Intention Estimation (PIE) [117], which include a large set of annotations.

Our research contributions presented in this chapter can be summarized as follows:

1. We extend the standard evaluation protocol to predict pedestrian crossing intentions as early as possible.
2. We build on top of a state-of-the-art action anticipation model by introducing a *goal* module to envision future motion in multiple feature spaces. We fuse these features with the motion history using an attention-based mechanism to predict crossing events.
3. We demonstrate, experimenting on multiple benchmarks, that pedestrian intents can be foreseen several seconds in advance, thus improving human safety and social awareness.

4.2 Intent Prediction Protocol

Pedestrian intent prediction relies on past motion to detect whether a pedestrian will cross the street or not, and it can be treated as a binary classification task at different anticipation times. In contrast to the standard benchmark proposed in [58], which predicts the crossing events in a time window between 1.0 and 2.0 seconds, our protocol anticipates the events as early as 4.0 seconds. The standard protocol, illustrated in Figure 4.2a, fixes the anticipation window in the $[1.0, 2.0]$ seconds for each pedestrian, with a fixed observation length of 0.5 seconds. To



(A) **Standard Protocol**: Anticipation range $[1.0, 2.0]$ seconds, with overlapped observations of 0.5 seconds.

(B) **Our Protocol**: Anticipation range $[0.1, 0.4]$ seconds, with adaptive observation window and simultaneous anticipations.

FIGURE 4.2: Standard intent prediction protocol [58] VS Our proposed protocol.

provide the predictions in the provided range, the protocol employs overlapped observations throughout the range, with a step of α seconds. Each extracted observation, paired with its corresponding anticipation, represents a distinct data point during both training and inference time (see $\left[P_1, P_2, \ldots P_{\frac{1}{\alpha}}\right]$ in Figure 4.2a).

On the other hand, our proposed protocol for intent prediction aims to provide earlier anticipations, utilizing the availability of longer records of the observed pedestrians, where it adaptively extends the observation window, given the targeted anticipation time. Furthermore, treating the overlapped observations as separate data points provides averaged prediction in the anticipation range, biasing the prediction of the model during early anticipations. Therefore, as we focus on earlier predictions, we overlooked the overlapped anticipations, where our proposed model produces simultaneous and separate predictions as the different anticipation times for each observed pedestrian, as shown in Figure 4.2b. This allows more accurate evaluation at the different anticipation times, especially at early anticipations. More specifically, let r be a set of RGB frames of a pedestrian starting from t_0 to the crossing event at time t_s, and let $t_s - t_a$ be the encoding time, where t_a represents the remaining time from the current observation until the event (anticipation time). Our task is to observe r from t_0 until $t_s - t_a$ and predict the crossing/not crossing action at t_s. We aim to predict crossing intentions at different anticipation times, from very early ($t_a \uparrow$) to very close to the event ($t_a \downarrow$).

4.3 RULSTM for Intent Prediction

As a backbone, we consider RU-LSTM [42], which processes the observed video frames in two stages: an encoding stage of S_{enc} steps and an anticipation stage of S_{ant} steps, with α as the time interval between subsequent time steps. As explained in Chapter 3, this model uses an LSTM-based encoder-decoder, referred to as *rolling*-LSTM (R-LSTM) and *unrolling*-LSTM (U-LSTM), respectively. During the decoding stage, it simultaneously produces predictions at various anticipation times, utilizing the binary cross entropy loss. RU-LSTM uses multi-modal features, where different modalities are fused using an attention-based mechanism[†](MATT).

4.4 Goal Module

4.4.1 Goal Estimation

Since RU-LSTM cannot use any information after $t_s - t_a$ during the evaluation phase, it repeats the observed frame at $t_s - t_a$ for each step in the unrolling stage. Nevertheless, this model may benefit from having a glimpse into the future. For this reason, we define a *goal* module that predicts features that might be extracted at t_s and fuse this information with the features at $t_s - t_a$ to enhance its unrolling capability. Let i be the index of the frame at time $t_s - t_a$; then, for each modality m, we use both encoding sequence (f^m_{enc}) and frame features (f^m_i) to predict future features at t_s, as follows:

$$G^m_i = D^m_{goal}(LSTM^m_{goal}(f^m_{enc}) \oplus f^m_i), \tag{4.1}$$

where $\oplus$ denotes the concatenation operation, $LSTM^m_{goal}$ represents the encoding process related to the goal prediction of modality m, and D^m_{goal} is a feed-forward

[†] We refer the reader to [42] for a full description of this model.

neural network. We employ the root mean square error to train our goal module, as given in (4.2).

$$L_{goal} = \sqrt{(G_i^m - f_{t_s}^m)^2} \tag{4.2}$$

4.4.2 Goal Fusion

Using an MLP-based network, we define an attention mechanism (see Fig. 4.3) to predict weights to be assigned to *goal* features at each unrolling stage, denoted by W_i^m in Eq. 4.3. W_i^m inherits its length l_i from the corresponding unrolling stage, where $l_i = \frac{t_a}{\alpha}$ is equal to the number of time steps in t_a seconds, with an interval of α seconds; for example, in Fig. 4.3, $l_0 = 5$, $l_1 = 4$, and the length keeps decreasing reaching the event, where $l_4 = 1$ at $t_a = \alpha$ and $t = t_s - \alpha$. Concretely,

$$W_i^m = Softmax(D_{GATT}^m(h_i^m \oplus c_i^m)) \tag{4.3}$$

where D_{GATT}^m is a feed-forward neural network representing our goal attention mechanism (GATT) which uses hidden (h_i^m) and cell (c_i^m) vectors of R-LSTM at time t_a. Our new input to the U-LSTM is then a weighted average of G_i^m and f_i^m, defined as $I_i^m = W_i^m \times G_i^m + (1 - W_i^m) \times f_i^m$.

Finally, predictions from each modality are combined with the modality attention (MATT) mechanism, as proposed in [42], to provide the final prediction.

4.5 Experimental Results

4.5.1 Datasets

As mentioned in Subsection 1.2.2, we rely on two popular datasets for our evaluation, namely JAAD and PIE. JAAD dataset contains 2786 pedestrian samples, annotated with bounding boxes, and is split into two subsets (JAAD$_{beh}$ and JAAD$_{all}$). PIE dataset contains 1842 annotated pedestrian samples with bounding boxes and behavioral tags, in addition to the ego vehicle's speed, GPS coordinates, and heading direction.

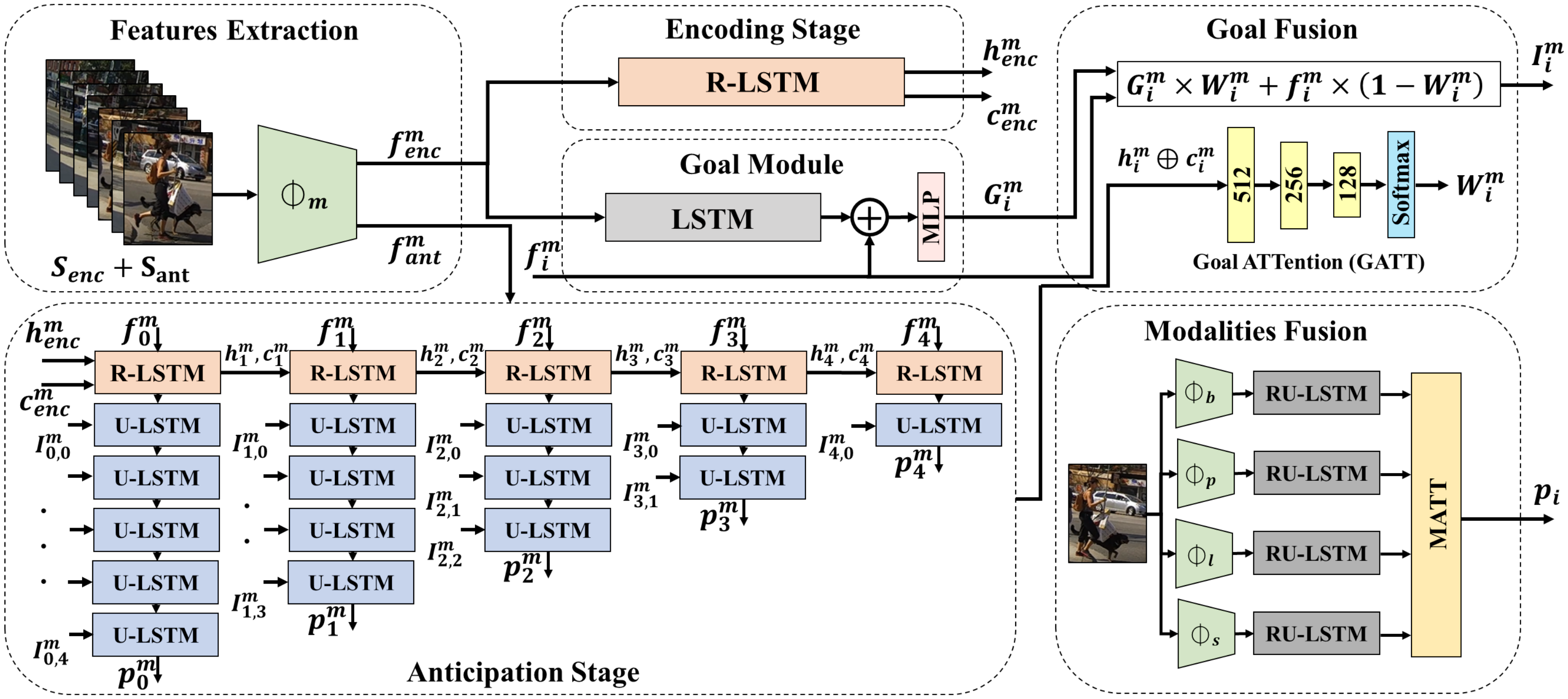

Features Extraction
$S_{enc} + S_{ant}$
Φ_m
f^m_{enc}
f^m_{ant}
Encoding Stage
R-LSTM
h^m_{enc}
c^m_{enc}
Goal Module
LSTM
MLP
G^m_i
f^m_i
Goal Fusion
$G^m_i \times W^m_i + f^m_i \times (1 - W^m_i)$
I^m_i
$h^m_i \oplus c^m_i$
512
256
128
Softmax
W^m_i
Goal ATTention (GATT)
Anticipation Stage
R-LSTM
U-LSTM
f^m_0 f^m_1 f^m_2 f^m_3 f^m_4
h^m_1, c^m_1 h^m_2, c^m_2 h^m_3, c^m_3 h^m_4, c^m_4
$I^m_{0,0}$ $I^m_{0,4}$ $I^m_{1,0}$ $I^m_{1,3}$ $I^m_{2,0}$ $I^m_{2,1}$ $I^m_{2,2}$ $I^m_{3,0}$ $I^m_{3,1}$ $I^m_{4,0}$
p^m_0 p^m_1 p^m_2 p^m_3 p^m_4
Modalities Fusion
Φ_b Φ_p Φ_l Φ_s
RU-LSTM
MATT
p_i

FIGURE 4.3: Our architecture is composed of six sequential steps: **1) Features Extraction**: where a function Φ_m extracts different types of features from raw images; **2) Encoding stage**: encodes the motion history with length S_{enc}; **3) Goal Module**: takes the input features of the encoding part, along with the features at the i^{th} anticipation step (f_i^m) and predicts goal features associated to the modality m (G_i^m); **4) Goal Fusion**: fuses predicted goal features G_i^m and extracted frame features f_i^m, and produces a new input representation for the next step; **5) Anticipation Stage**: takes I_i^m as input and predicts the crossing probability (binary crossing/not crossing events) at each time step (p_i^m); **6) Modalities Fusion**: represents our final stage taking predictions provided by each modality and applies an attention-based fusion mechanism to produce the final prediction.

Three main modalities for both datasets are employed: bounding box coordinates, pose features, and visual local context features. The PIE dataset also includes ego-vehicle speed as an additional modality.

4.5.2 Implementation Details

Our LSTMs have 512 hidden layers, with $S_{enc} = 6$ and $S_{ant} = 40$. The time interval is 3 frames ($\alpha = 0.1$ s), while our models output 40 predictions from $t_a = 4$ s until $t_a = 0.1$ s. For the sake of clarity, we consider only predictions at $t_a \in \{4, 3, 2, 1\}$ s. For each anticipation time (t_a), we discard videos that do not satisfy the minimum length requirement, *i.e.*, videos whose length is less than t_a. Therefore, we consider 4 s as the earliest anticipation time, where longer predictions would lead to discarding too many examples (more than 50% of data from the JAAD dataset). For a fair comparison, in the range [2-1] s, we use the same evaluation protocol proposed in [58], averaging the predictions with a step of 0.1 s for JAAD and 0.2 s for PIE.

4.6 Baselines

To evaluate the performance of the proposed method for early predictions, we develop an LSTM-based model using RU-LSTM without the unrolling stage (R-LSTM). Furthermore, we compare our model, at different anticipation times, against PCPA [58], which uses fixed and overlapped observations of 0.5 s in the range $[2-1]$ s earlier the event. This model is modified to output earlier predictions in the range $[4-1]$ s. We also consider CAPformer [68] and TrouSPI-Net [70] models.

4.6.1 Results and Baselines Comparison

Firstly, we evaluate our models on different anticipation times and, eventually, measure the impact of the *goal* module. Table 4.1 reports our results for different anticipation times (from 4 s to 1 s). Depending on the dataset, two main trends emerge when approaching the event. On the JAAD$_{beh}$ dataset, all metrics remain stable approaching the event, confirming our intuition to forecast features in advance. This behavior may also be related to the limited samples of this dataset, which is quite unbalanced, yet on JAAD$_{all}$, this trend is confirmed since no remarkable drops in performance can be observed. Among the used models,

	JAAD$_{beh}$														
	4 s			3 s			2 s			1 s			[2-1] s		
	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
PCPA [58]	-	-	-	-	-	-	-	-	-	-	-	-	0.58	0.50	0.71
PCPA† [58]	0.54	0.51	0.62	0.47	0.46	0.54	0.49	0.45	0.61	0.45	0.52	0.63	0.56	0.50	0.67
R-LSTM	0.67	0.64	0.74	0.70	0.64	0.78	0.66	**0.62**	0.75	0.65	0.60	0.75	0.65	0.59	0.74
RU-LSTM	**0.72**	**0.67**	**0.79**	**0.72**	**0.64**	**0.81**	**0.69**	**0.62**	**0.78**	**0.70**	**0.63**	**0.79**	**0.69**	**0.62**	**0.78**
	JAAD$_{all}$														
PCPA [58]	-	-	-	-	-	-	-	-	-	-	-	-	0.85	**0.86**	**0.68**
PCPA† [58]	0.75	0.75	0.50	0.74	0.76	0.53	0.72	0.75	0.52	0.76	**0.79**	0.55	0.80	0.79	0.57
R-LSTM	0.83	0.74	0.54	0.86	0.73	0.58	**0.85**	0.73	0.57	**0.87**	0.77	**0.62**	**0.86**	0.76	0.60
RU-LSTM	**0.84**	**0.76**	**0.57**	**0.87**	**0.78**	**0.64**	**0.85**	**0.76**	**0.59**	0.86	0.78	**0.62**	**0.86**	0.78	0.62
	PIE														
PCPA [58]	-	-	-	-	-	-	-	-	-	-	-	-	**0.87**	**0.86**	**0.77**
PCPA† [58]	0.76	0.75	0.62	0.77	0.76	0.63	0.83	**0.84**	0.73	0.86	**0.85**	0.77	0.86	**0.86**	**0.77**
R-LSTM	0.75	0.64	0.48	0.75	0.66	0.50	0.76	0.66	0.51	0.76	0.67	0.52	0.76	0.67	0.52
RU-LSTM	**0.77**	**0.76**	**0.63**	**0.80**	**0.79**	**0.68**	**0.85**	0.82	**0.74**	**0.88**	**0.85**	**0.79**	**0.87**	0.84	**0.77**

Table 4.1: Comparison among multiple intent prediction models for both JAAD and PIE datasets. We consider four anticipation times and the standard evaluation protocol averaging predictions within the range [2-1] s. PCPA† [58] denotes our retrained PCPA model (using the original code and configurations provided in the official GitHub repository).

RU-LSTM performs the best in both cases. By contrast, on the PIE dataset, our metrics increase when the anticipation time gets close to the event. It is worth noting that RU-LSTM systematically improves performance metrics compared to the other considered models for the different anticipation times. In the $[2-1]$ s range, RU-LSTM outperforms the considered models on JAAD$_{beh}$ dataset while is on par with PCPA [58] on JAAD$_{all}$ and PIE datasets. It is also worth mentioning that the standard protocol used in [58] increases the number of training samples considering overlapping windows. By contrast, our proposed protocol dumps this overlapping technique, leading to a more realistic evaluation of this task yet largely reducing the number of training samples. This could also explain the limited performance of RU-LSTM in the $[2-1]$ s range, where PCPA uses overlapping windows. Meanwhile, RU-LSTM outperforms all the models using the same number of training samples at fixed anticipation times.

Table 4.2 compares our model to state-of-the-art architectures. RU-LSTM does not contain the goal module, while G-RULSM uses future features estimation. We observe that G-RULSTM outperforms the state-of-the-art models for JAAD$_{beh}$, in addition to a noticeable improvement over RU-LSTM. For JAAD$_{all}$, goal-boosted

	JAAD$_{beh}$			**JAAD$_{all}$**		
	Acc	AUC	F1	Acc	AUC	F1
PCPA [58]	0.58	0.50	0.71	0.85	**0.86**	**0.68**
CAPformer [68]	-	0.55	0.76	-	0.82	0.63
TrouSPI-Net [70]	0.64	0.56	0.76	0.82	0.77	0.58
RU-LSTM	0.69	0.62	0.78	**0.86**	0.78	0.62
G-RULSTM	**0.72**	**0.65**	**0.80**	**0.86**	0.80	0.63
Imp.	**3%**	**3%**	**2%**	-	**2%**	**1%**

Table 4.2: Comparison between our architectures and state-of-the-art models within the $[2-1]$ s range on JAAD dataset.

	JAAD$_{beh}$				
	4 s	3 s	2 s	1 s	[2-1] s
			Bounding Box		
Without Goal	0.55	0.59	0.63	0.64	0.58
Interpolation	**0.61**	0.56	0.61	**0.65**	**0.64**
Concatenation	**0.61**	**0.64**	0.63	0.63	0.63
Attention	**0.61**	0.63	**0.65**	**0.65**	**0.64**
			Pose		
Without Goal	0.51	0.60	0.65	0.65	0.65
Interpolation	0.59	0.64	0.63	0.64	0.63
Concatenation	0.60	**0.67**	**0.66**	**0.66**	**0.66**
Attention	**0.62**	**0.67**	0.57	0.65	0.65
			Local Context		
Without Goal	0.68	0.66	0.65	0.67	0.66
Interpolation	0.73	0.67	0.63	0.66	0.61
Concatenation	0.65	**0.70**	**0.68**	0.68	**0.68**
Attention	**0.69**	0.67	0.66	**0.69**	0.66

Table 4.3: Ablation study reporting the accuracy metric using different fusion methods for each modality on JAAD$_{beh}$. Underlined numbers refer to the 2^{nd}-best-performing model.

G-RULSTM outperforms our baseline for AUC and F1 metrics, which are more robust metrics for unbalanced datasets. Our proposed model suffers from a hard reduction of training samples, compared to [58], which limits its performance on larger datasets, *e.g.*, JAAD$_{all}$. Nevertheless, our model is on par with CAPformer [68] and outperforms TrouSPI-Net [70] for both subsets.

4.6.2 Ablation Study

Table 4.3 reports an ablation study considering three different fusion techniques for estimating future features: **1) Interpolation**, where I_i^m is obtained as a linear combination of f_i^m and G_i^m, based on the time distance d_j from the current step to the frame corresponding to the event, i.e, $I_i^m = \frac{(1-d_j)}{l_i} \times G_i^m + \frac{d_j}{l_i} \times f_i^m$; **2) Concatenation** followed by an MLP layer; **3) Attention**, as defined in Sec. 4.4. We observe that our goal module, with any considered fusion technique, improves prediction metrics over the baseline for all modalities and anticipation times. Furthermore, a different fusion technique could be considered depending on the considered modality. For example, for bounding box coordinates representing the pedestrian's motion within the image, a linear relationship over time is noticed. By contrast, both pose and local context features show a different trend. In this case, concatenation and attention perform better. We select the attention-based technique for all modalities to adapt to both their linear and non-linear relationships.

4.7 Conclusion

In this chapter of our research, we revise the standard evaluation protocol used to measure the performance of pedestrian intent prediction models. We demonstrate that crossing events can be predicted several seconds in advance with no (or negligible) impact on the performance. To validate our intuition, we build upon an action anticipation model, a *goal* module to forecast future features and improve its prediction metrics. This information can increase prediction accuracy up to 3% compared to models that do not envision future features.

Chapter 5

TAMFORMER: Temporal Adaptive Mask Transformer for Early Intent Prediction

Chapter Abstract

This chapter focuses on our TAMFORMER model, with ablation and evaluation. We propose a novel approach based on a multi-modal transformer. Our model encodes past observations and produces multiple predictions at different anticipation times. Moreover, we propose to learn the attention masks of our transformer-based model in order to weigh differently present and past temporal dependencies. We investigate our method on several public benchmarks for early intention prediction, improving the prediction performances at different anticipation times. The work in this chapter has been published in [52]. [*]

5.1 Introduction

In this chapter, we continue in the pedestrians' early intention prediction domain, in which, from a current observation of an urban scene, the model predicts the future crossing/not crossing actions of pedestrians approach the street. Our method is based on a multi-modal transformer that encodes past observations and produces

[*] This chapter has been published as "TAMFORMER: Multi-Modal Transformer with Learned Attention Mask for Early Intent Prediction" in *Proc. of IEEE ICASSP International Conference on Acoustics, Speech, and Signal Processing, 2023.*

multiple predictions at different anticipation times. Moreover, we propose to learn the attention masks of our transformer-based model (**T**emporal **A**daptive **M**ask Trans**former**) in order to weigh differently present and past temporal dependencies. We investigate our method on several public benchmarks for early intention prediction, improving the prediction performances at different anticipation times compared to the previous works. In particular:

1. We propose a new model for early intent prediction based on a multi-modal multi-predictions transformer.
2. We propose a new mechanism for learning adaptive attention masks inside the transformer, leading to better performances and more efficient computation.
3. We propose a novel regularization loss function to improve the early predictions.
4. We propose a data augmentation technique to overcome the problem of limited training data.
5. We conduct several experiments and model ablations on different datasets, obtaining state-of-the-art results on the early intent prediction task.

5.2 Multi-Modal Transformer

Our proposed TAMformer model, depicted in Fig. 5.1, has three major components: the *Encoding* in which the multi-modal input is encoded, the *Query* where the future query is built, and the *Decoding* where the future predictions are computed.

5.2.1 Value Encoding

Raw images are projected to different modalities with Φ_m, where $m \in \{1, \dots, M\}$, to $\mathbf{x}_m \in \mathbb{R}^{T \times D_m}$ and subsequently passed to a transformer block TE_m that creates an encoded representation $\mathbf{z}_m \in \mathbb{R}^{T \times D_m}$, where T is the time sequence length and D_m is the feature size:

$$\mathbf{z}_m = TE_m(\mathbf{x}_m + \mathbf{p}), \quad \mathbf{z}_e = Cat\Big[\mathbf{z}_1, \dots, \mathbf{z}_M\Big].$$

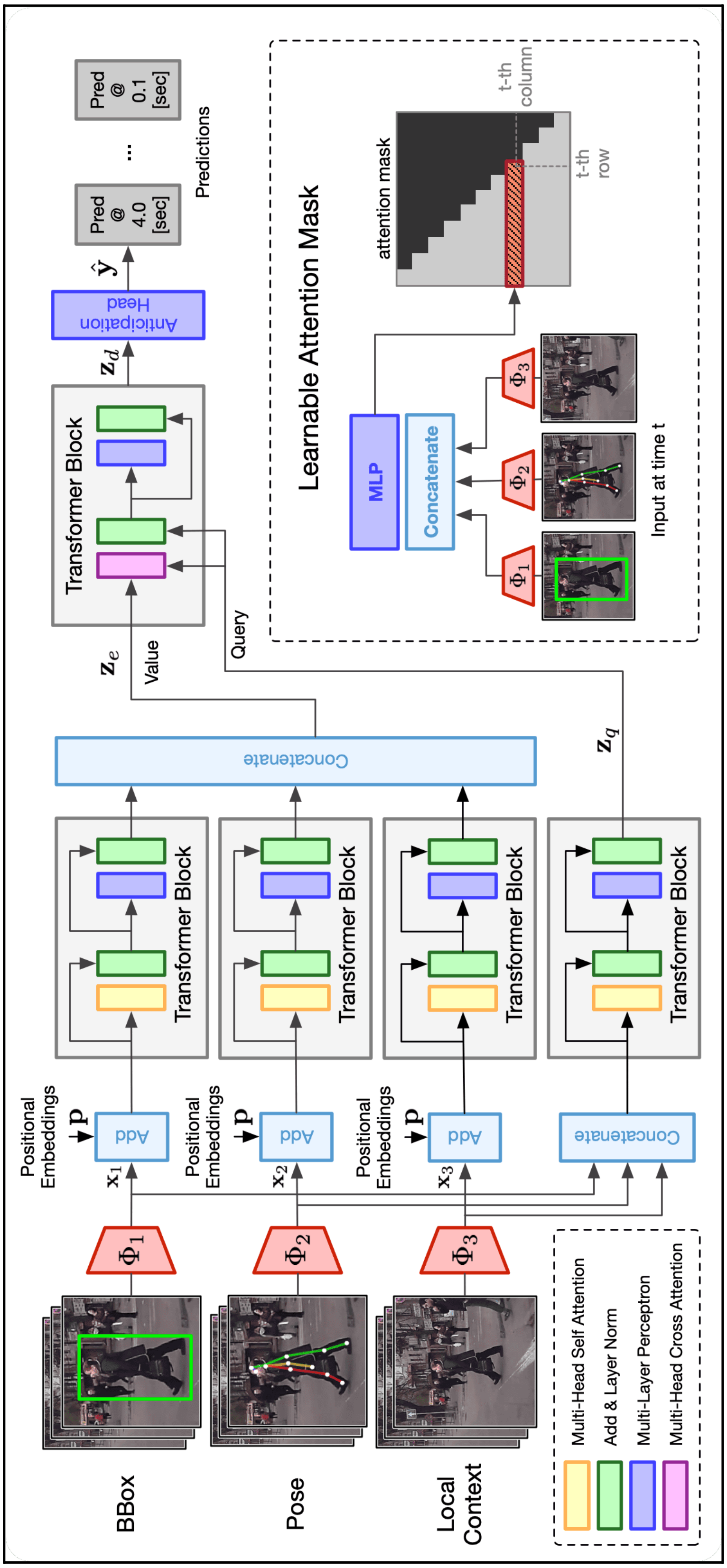


Figure 5.1: TAMformer model architecture.

In order to preserve the order of the sequence, the positional encodings $\mathbf{p}$ are added to the input sequence after the linear projection.

Our model does not assume any particular input modalities; however, we used the RGB local context, bounding box coordinates, pose, and the ego vehicle speed in our work.

5.2.2 Query Encoding

Instead of applying a single late fusion of the encoded sequences, we allow the input features to interact in an early fusion step, allowing to maximize the extracted information during the fusion process from both the row modalities and their encoded representations. A transformer block TQ processes the concatenated features, creating a query at each time step, as follows:

$$\tilde{\mathbf{z}}_q = Cat\Big[\mathbf{x}_1, \ldots, \mathbf{x}_M\Big], \quad \mathbf{z}_q = TQ(\tilde{\mathbf{z}}_q)$$

5.2.3 Decoding and Anticipation

A transformer decoder block TD processes the encoded representation $\mathbf{z}_e$ and, for each query in $\mathbf{z}_q$ produces a decoded representation through the cross-attention mechanism that subsequently is projected to the final prediction as follows:

$$\mathbf{z}_d = TD(\mathbf{z}_e, \mathbf{z}_q), \quad \hat{\mathbf{y}} = Sigmoid(MLP(\mathbf{z}_d))$$

5.3 Temporal Adaptive Mask

Usually, video frames are redundant when processed at a high frame rate, and, by contrast, at a low frame rate, the information can be lost as the sampling does not consider frame importance. For these reasons, we propose a method that allows the model to choose the frames that maximize the information and minimize redundancy. As depicted in Fig. 5.1, at the t-th step, the input features are concatenated and fed to a feed-forward network that outputs a learned mask $\mathbf{M}_t$. We decided to encode the representations at a full frame rate (30 FPS) and to make predictions at a sub-sampled frame rate (10 FPS) for more efficient computation. In our model, we have two types of masks $\mathbf{M}_e$, $\mathbf{M}_d$ related to the encoding and decoding transformer blocks and, in order to avoid future information conditions in the present prediction, the masks are causal, and their t-th rows are predicted as follows:

$$\tilde{\mathbf{z}} = Cat\Big[\mathbf{x}_1, \dots, \mathbf{x}_M\Big],$$
$$\mathbf{M}_{[:t]} = Sigmoid(MLP(\tilde{\mathbf{z}}_{[:t]})).$$

5.4 Auxiliary Loss

Typically, anticipation models perform better as they get closer to the anticipated action. Thus, we propose an auxiliary regularization loss function:

$$\mathcal{L}_r = \sum_t \|\mathbf{z}_d[t] - \mathbf{z}_d[T]\|^2$$

that minimizes the gap between the current decoder embedding $\mathbf{z}_d[t]$ and the final one $\mathbf{z}_d[T]$. We found beneficial to train the model in two stages: we first pre-train the system using only the cross-entropy loss $\mathcal{L}_{ce}$ for action anticipation, and subsequently we add the regularization term $\mathcal{L}_r$ to the total loss ($\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_r$), encouraging the earlier anticipation predictions to benefit from the last decoder representation that can observe the whole sequence before the action starts.

5.5 Data Augmentation

In contrast to the standard protocol, [118], we abandon overlapped samples and follow the proposed protocol in [51], treating each pedestrian as a single sample. Consequently, a hard reduction in the number of samples is present, compared to [118]. However, transformers require large training data for the best results. Accordingly, we propose a data augmentation procedure to increase the training data. As in [51], the observation length is 4.5s, ignoring any earlier frames in the sample. We benefit from such frames to augment the samples, replacing the encoding window with earlier frames when they exist. Thus, more versions of the same sample with different encoding windows are available.

5.6 Experimental Results

5.6.1 Implementation Details

We rely on the urban-scenarios datasets used in Chapter 4, with the same anticipation protocol and using the standard classification metrics for evaluation: Accuracy, AUC, and F1-Score. The training procedure includes two phases (500 epochs each): a pre-training phase on action anticipation and a tuning phase with the regularizer $\mathcal{L}_r$. We used the SGD optimizer with learning rates $lr = \{10^{-5}, 10^{-2}, 10^{-3}\}$ for PIE, JAAD$_{all}$, and JAAD$_{beh}$ respectively. Each transformer block has $N_h = 6$ heads, $ff_{dim} = 1024$, and the MLP producing the learned masks consists of $N_l = 3$ layers with sizes $\{128, 64, 32\}$.

5.6.2 Quantitative Results

We compare our model with PCPA [118], which represents the SOTA work in intent prediction and an adapted PCPA version that can produce earlier anticipations. Although we are not applying the overlapping protocol in [118], we align with it on the used samples and anticipation range during evaluation to allow for a fair comparison. Additionally, we compare with a single LSTM (R-LSTM), RULSTM [119], and G-RULSTM [51]. Following [118], Table 5.1 reports the comparison in the anticipation range of $[2-1]$ s, and the main architecture differences. We observe an F1-score out-performance gap that reaches +2% on PIE and +5% on JAAD$_{all}$, comparing our TAMformer to the best model in the table. Moreover, we reported a comparison on different anticipation times from 4s to 1s in Table 5.2 and, depending on the dataset, we notice two trends: for PIE, TAMformer outperforms by almost +2% on F1-score in all anticipation times. Nevertheless, on JAAD, our model suffers a degraded performance at early anticipation ($[4-3]$s) while maintaining the improvements on JAAD$_{all}$ (maximum +9%) and on JAAD$_{beh}$ (maximum +2%). The reduction in training samples in early anticipation ($> 50\%$ on JAAD) could explain this degradation, as transformers need lots of training samples.

	Visual Backbone	Blocks	Fusion	t_a	S_{temp}	**PIE**			**JAAD$_{all}$**			**JAAD$_{beh}$**		
						Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
PCPA† [118]	C3D	GRU	L-ATT	$[2-1]$s	30 FPS	0.86	**0.86**	0.77	0.8	0.79	0.57	0.56	0.5	0.67
R-LSTM [119]	VGG16	LSTM	L-ATT	$[4-0.1]$s	10 FPS	0.76	0.67	0.52	0.86	0.76	0.6	0.65	0.59	0.74
RU-LSTM [119]						0.87	0.84	0.77	0.86	0.78	0.62	0.69	0.62	0.78
G-RULSTM [51]						-	-	-	0.86	0.8	0.63	0.72	0.65	0.8
TAMformer (ours)	VGG16	TF	EC+LC	$[4-0.1]$s	Adaptive	**0.88**	**0.86**	**0.79**	**0.88**	**0.83**	**0.68**	**0.73**	**0.69**	**0.8**

Table 5.1: Architectural and performance Comparison of different SOTA models in the standard anticipation range $[2-1]$s. We compare three architectural aspects: 1) **Visual Backbone**, the backbone model used in extracting context features from images; 2) **Blocks**, the time sequence processing framework; 3) **Fusion**, the merging technique of the multi-modalities, where **L-ATT** stands for **L**ate **ATT**ention, **EC** is **E**arly **C**oncatenation, and **LC** is **L**ate **C**oncatenation.

	PIE											
	4 s			3 s			2 s			1 s		
	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
PCPA† [118]	0.76	0.75	0.62	0.77	0.76	0.63	0.83	0.84	0.73	0.86	0.85	0.77
R-LSTM [119]	0.75	0.64	0.48	0.75	0.66	0.50	0.76	0.66	0.51	0.76	0.67	0.52
RU-LSTM [119]	0.77	0.76	0.63	0.80	0.79	0.68	0.85	0.82	0.74	**0.88**	0.85	0.79
TAMformer (ours)	**0.78**	**0.77**	**0.65**	**0.81**	**0.81**	**0.7**	**0.87**	**0.84**	**0.76**	**0.88**	**0.88**	**0.8**
	JAAD$_{all}$											
PCPA† [118]	0.75	0.75	0.50	0.74	0.76	0.53	0.72	0.75	0.52	0.76	0.79	0.55
R-LSTM [119]	0.83	0.74	0.54	0.86	0.73	0.58	0.85	0.73	0.57	0.87	0.77	0.62
RU-LSTM [119]	0.84	**0.76**	**0.57**	**0.87**	0.78	**0.64**	0.85	0.76	0.59	0.86	0.78	0.62
TAMformer (ours)	**0.85**	0.75	0.56	0.86	**0.79**	**0.64**	**0.89**	**0.82**	**0.68**	**0.89**	**0.82**	**0.7**
	JAAD$_{beh}$											
PCPA† [118]	0.54	0.51	0.62	0.47	0.46	0.54	0.49	0.45	0.61	0.45	0.52	0.63
R-LSTM [119]	0.67	0.64	0.74	0.70	0.64	0.78	0.66	0.62	0.75	0.65	0.60	0.75
RU-LSTM [119]	**0.72**	**0.67**	**0.79**	0.72	0.64	**0.81**	0.69	0.62	0.78	0.70	0.63	0.79
TAMformer (ours)	0.68	0.62	0.77	**0.73**	**0.68**	0.80	**0.73**	**0.7**	**0.79**	**0.74**	**0.69**	**0.81**

Table 5.2: Performance at different anticipation times $[4-1]$s

5.6.3 Ablation Study

5.6.3.1 Time Scale Modeling

In Table 5.3, we evaluate the effect of processing input at different time scales in the model. Three approaches are tested: single and fixed scales (30 FPS and 10 FPS), multi-scale (SlowFast [10 FPS-30 FPS]), and our adaptive scale. As

	JAAD$_{all}$			**JAAD**$_{beh}$		
	Acc	AUC	F1	Acc	AUC	F1
30 FPS	0.84	0.78	0.6	0.63	0.51	0.77
10 FPS	0.86	**0.79**	0.63	0.64	0.56	0.76
SlowFast	**0.88**	0.77	0.64	**0.67**	**0.62**	0.75
Adaptive	0.87	0.78	**0.64**	**0.67**	0.58	**0.78**

TABLE 5.3: Effect of Time Scale

noticed, scaling down can improve performance by discarding much redundant information. Almost better performance can be achieved by applying the SlowFast multi-scaling that allows the model to benefit better from all available information. Yet, allowing the model to choose where to look should be the best option concerning the reported results.

5.6.3.2 Model Variants

Table 5.4 compares the model's different variants, where the best performance is triggered by increasing the training samples and applying the $\mathcal{L}_r$ loss. Additionally, Fig. 5.2 illustrates the effect of applying the $\mathcal{L}_r$ loss on all anticipation times, where a noticeable increase in the F1-score is present, especially at early anticipation times on the JAAD dataset.

TAS	DI	$\mathcal{L}_r$	**JAAD**$_{all}$			**JAAD**$_{beh}$			**PIE**		
			Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
✓	✗	✗	0.87	0.78	0.64	0.67	0.58	0.78	**0.88**	0.85	0.78
✓	✓	✗	**0.88**	0.79	0.65	0.69	0.68	0.75	-	-	-
✓	✓	✓	**0.88**	**0.83**	**0.68**	**0.73**	**0.69**	**0.8**	**0.88**	**0.86**	**0.79**

TABLE 5.4: Models Variants (**DI** stands for Data Increase)

5.6.4 Qualitative Results

Fig. 5.3 is an example of a learned mask and the corresponding input images. For illustration, only the mask corresponding to the first 0.5s of observation is shown.

[†] The reported results are our run of PCPA. As reported on GitHub, there are issues on reproducing the results of the original paper from the code.

The model chooses a different set of history frames at each time step that should maximize the information and minimize the redundancy at the corresponding time step. For example, at $t_a = 4$s, the model uses only 5 frames from the available 16 frames. Given the raw images, we observe much redundancy, yet some differences in the chosen images by the model.

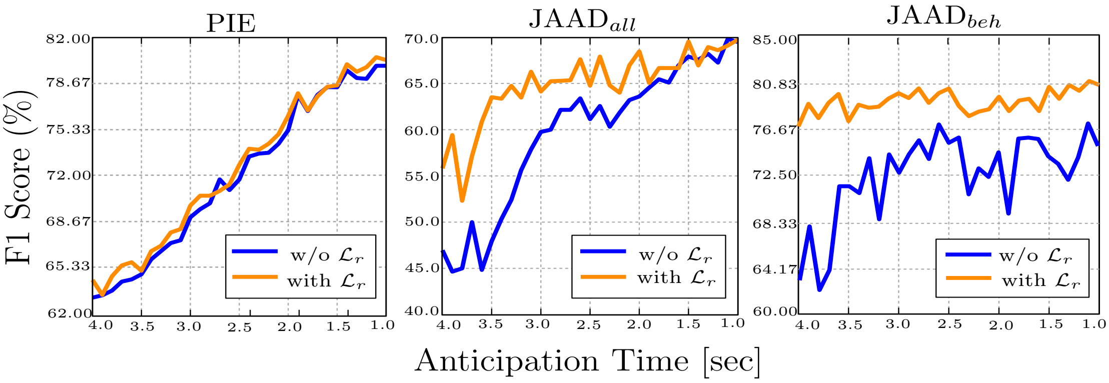


FIGURE 5.2: Effect of using the $\mathcal{L}_r$ loss.

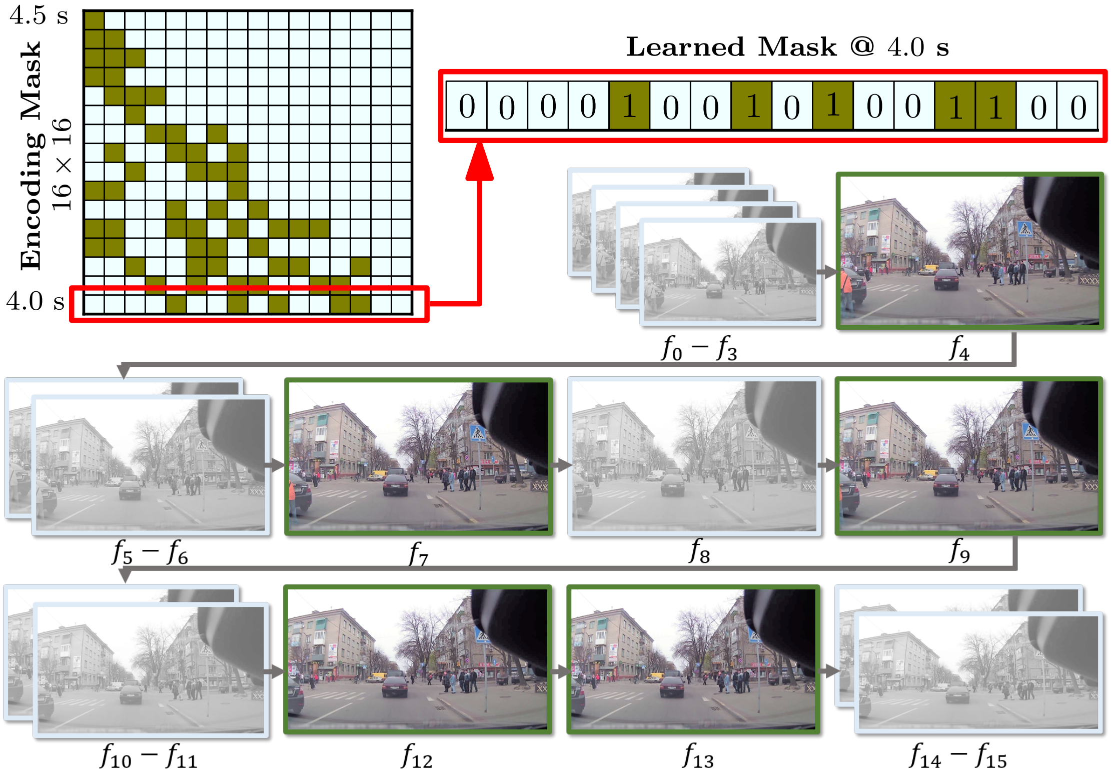


FIGURE 5.3: Qualitative example of a 0.5 s part of a learned attention mask, starting at $t_a = 4.5$ s until $t_a = 4.0$ s, where t_a is the anticipation time.

5.7 Conclusion

In this chapter of our work, we propose a multi-modality transformer-based model that adaptively learns attention masks to measure the temporal sequence's correspondences. We applied a new loss function to minimize the gap in performance between early anticipation times and the closest one to the anticipated action. The experiments demonstrate the proposed model's out-performance, which can reach +2% F1 on PIE and +5% F1 on JAAD, in the $[2-1]$s range. Similarly, TAMformer surpasses at early anticipation times, mainly on PIE. Yet, our model suffers a drop in performance at early anticipation times on JAAD. Thus, our future work will focus on achieving robust performance at all anticipated times with possible enhancement through adding or enhancing the used modalities, for example, by considering more modern context feature extraction with Residual networks or the ViT model. Additionally, more modalities could add value to the anticipation, such as the inclusion of point cloud features or textual features

Chapter 6

Language-Aided Action Anticipation with TAMFORMER

Chapter Abstract

This chapter explains the language-aided action anticipation part of our research. We investigate the effect of taking advantage of a language modality in pedestrian action anticipation, studying various captioning techniques of the observed frames, and integrating the generated text into our TAMFORMER model. Additionally, we expand the binary crossing/not crossing pedestrian action anticipation into multi-action anticipation. Experimental validation of our techniques on a large-scale dataset (LOKI) proves the notable effectiveness of including text in increasing the model comprehension and, consequently, increasing the performance. [*]

6.1 Introduction

Language is a multifaceted tool for enriching visual content, offering a comprehensive and contextual means to describe the observed scenes. Its capacity to furnish supplementary context and intricate details exceeds the details derived from visual feature extraction alone. Thanks to the witnessed advances in natural language processing (NLP), vision-language coupling is becoming more applicable, adding noticeable value to visual tasks. Notably, contemporary models like CLIP

[*] The work presented in this chapter is still in-progress for publication.

[101] provide an opportunity for enhancing visual tasks through integrated textual input.

In action anticipation, recent models have introduced innovative approaches, incorporating the generation of textual labels and seamlessly integrating them into anticipation models [105–109]. However, this fusion of vision and language encounters the challenge of providing textual ground truth, especially in complicated environments like urban scenes. In this part of our work, we integrate language into the pedestrian action anticipation task, where the decisions of pedestrians are affected by multiple factors, including the taken action by the person, the demographic profile, the collective behavior of surrounding people, and the overall state of the street. Thus, leveraging language allows for a more comprehensive understanding of these diverse factors compared to relying solely on visually extracted representations. Therefore, We dive into the task of generating informative textual descriptions for the input images. In particular, we can summarize our contributions in this part as follows:

1. We extend the pedestrian action anticipation task to anticipating multiple actions instead of the conventional crossing/not crossing prediction task.
2. We integrate vision and language to enhance the performance of action anticipation in complex urban scenarios.
3. We conduct an in-depth exploration of various techniques for generating per-frame textual captions.
4. We evaluate our approach on a novel large-scale urban dataset, validating the effectiveness of vision-language coupling in action anticipation tasks.

6.2 Language TAMFORMER

Our TAMFORMER model (see chapter 5) is designed to support multi-modalities. Therefore, we investigate the influence of integrating vision and language modalities in TAMFORMER. As depicted in Figure 6.1, the textual input is added as a novel modality during the encoding process, value encoding, and query encoding, in addition to being used as input in the learned mask module. The text description is set per frame to catch the progressing behavior of the person, along with the possible changes in the context.

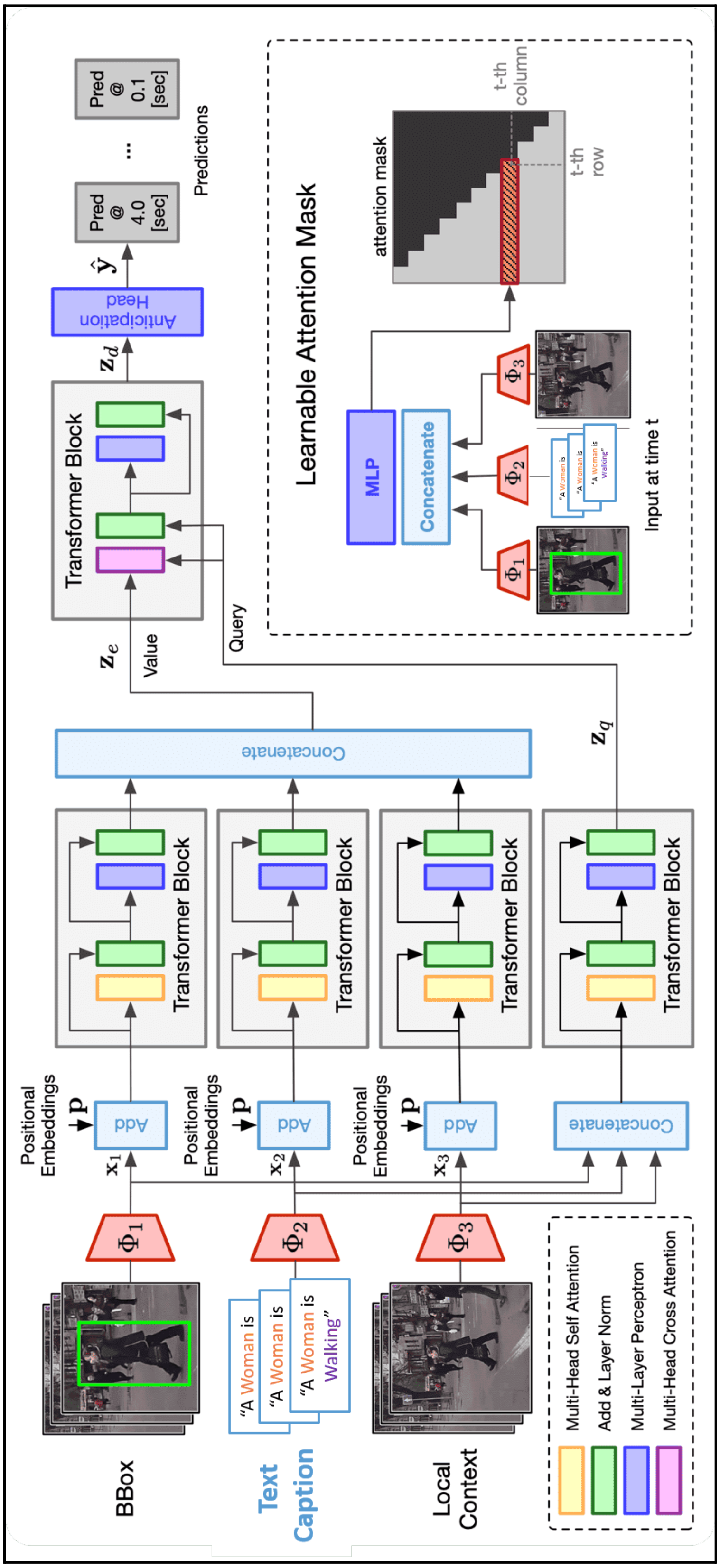


Figure 6.1: Language TAMformer

The study applied in [109] confirmed the effectiveness of the pre-trained CLIP framework [101] in representing the raw images and extracting meaningful information with respect to action anticipation. Relying on that, we use a pre-trained CLIP model to project the raw images and textual descriptions into the feature space. For each modality m, an Φ_m function is applied to project the raw input, where $\Phi_{img} = CLIP_{img}$ and $\Phi_{txt} = CLIP_{txt}$. $CLIP_{img}$ and $CLIP_{txt}$ represent the encoded representation extracted from a pre-trained CLIP model for images and captions, respectively.

6.3 Captioning Techniques

The main challenge in integrating language into the pedestrian action anticipation task is the absence of textual labeling for the images or the scene. In our work, we investigate different techniques for generating textual labels for the images.

6.3.1 Predefined Captions

As our task addresses action anticipation in urban scenarios, this could limit the range of possible activities that can be performed in an urban environment. A straightforward technique to couple an image with a textual description is to predefined a set of likely descriptions and then pair a given image with its closest description in feature space. We split a textual description into four parts: Demographic description, basic activity description, behavioral description, location description, and interaction description. Table 6.1 summarizes the possible descriptions in each category, where each category aims to cover a possible aspect in a given scene. The final list of predefined captions represents all possible combinations between the defined text in all categories. The chosen caption C_I for an image I should have the closest representation to the image, as in (6.1), where K is the number of predefined captions.

$$C_I = \arg\min_{k=1}^{K} \| CLIP_{img}(I) - CLIP_{txt}(C_k) \| \tag{6.1}$$

6.3.2 Image Captioning

A more reasonable approach for getting a textual description for an image is using an image captioner. BLIP-2 [104] is a recent image captioning, producing robust

Description Category	List of Predefined Captions
Demographic Description	A woman is A man is A girl is A boy is
Basic Activity	walking - standing
Location Description	on the sidewalk on zebra line next to a bus station next to a car next to traffic lights in front of a building
Behavioral Description	while talking to someone while talking in the phone while looking at the phone and raising his hand while carrying a baby while pushing a strolling holding a crutch
Interaction Description	with a child with a group of people with a dog
ex. A man is walking on the sidewalk *ex.* A woman is standing in front of a building while talking in the phone with a child	

TABLE 6.1: Predefined Captions

image captions in multiple domains. We rely on a pre-trained BLIP-2 model to generate our per-frame captions. However, many factors can affect the generated captions, such as the scale of the context or the amount of noise in the image. Therefore, we conduct multiple steps to ensure the accuracy and clarity of the captions, including multi-spatial-scale captioning and caption cleaning.

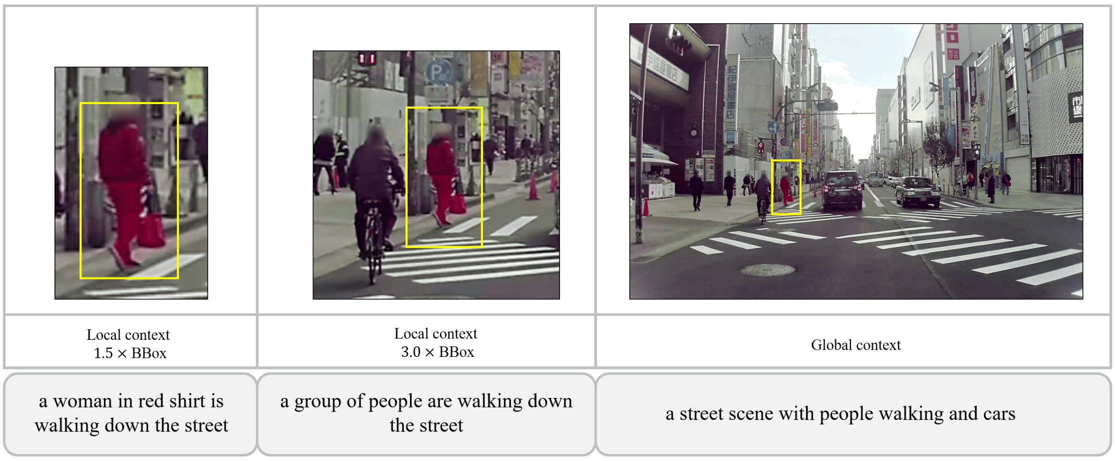

Local context
1.5 × BBox
Local context
3.0 × BBox
Global context
a woman in red shirt is walking down the street
a group of people are walking down the street
a street scene with people walking and cars

FIGURE 6.2: An example of the generated captions using different cropping scales around the pedestrian in question. The **Global Context** is the complete scene, without cropping. While a **Local Context** means a cropped image with a ratio/scale of the size of the pedestrian bounding box (*BBox*).

6.3.2.1 Multi-Spatial-Scale Captioning

A single frame in our input represents a wide-view urban scene; however, as we focus on a single pedestrian in the scene, the frame is cropped, centering the pedestrian in the cropped image while adding a reasonable amount of context to the image. However, the amount of context added to the cropped image affects the interpretation of the image and, consequently, the generated caption, as illustrated in Figure 6.2. To allow more informative captions, we employ different cropping scales, capturing different contexts and information from the scene. Let S be the number of considered cropping scales, and C^s_{cap} is the description generated by the captioner for scale s, then the feature representation $\mathbf{x}_{cap\_txt}$ is given by (6.2).

$$\mathbf{x}_{cap\_txt} = \sum_{s=1}^{S} CLIP_{txt}(C^s_{cap}) \tag{6.2}$$

6.3.2.2 Captions Refinement

The generated captions incur a large amount of noise. Firstly, the pre-trained model (BLIP-2) is not specialized in urban scenarios; therefore, the captions could contain unrelated descriptions of an urban environment. Then, we have a high percentage of noisy images due to zooming in for centering a relatively far pedestrian, leading to corrupted captions, as exemplified in Figure 6.3. To overcome

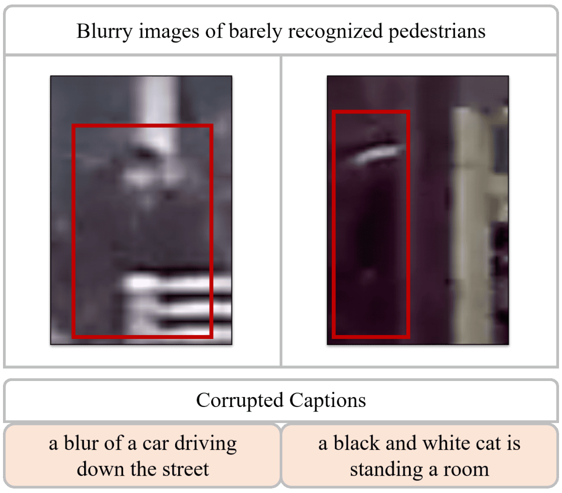

Blurry images of barely recognized pedestrians
Corrupted Captions
a blur of a car driving down the street
a black and white cat is standing a room

FIGURE 6.3: Examples of corrupted captions due to unclear and noisy images.

such noisy captions, we create a dictionary of words related to the urban environment and activities, *e.g., street, car, walk, etc.* Then, we ignore any generated caption that does not describe a street scene.

6.3.3 Template-based Captioning

A more stable technique for getting text captions for images could be achieved through a fixed text template filled with correct information. Still, this technique is conditioned on the availability of such information. In our work, we examine this text-generation technique in our setup, where the text template *"A/An **person_des** is **action_des**"* is used. The keyword "***person_des***" is replaced with the corresponding pedestrian description (*age/gender*), and "***action_des***" is replaced by the detected activity in the corresponding frame, *i.e., walking, standing, crossing, etc.* For example, one description could be "*An adult female is walking in the street*".

The chosen attributes reflect the most important factors affecting the future activity of the pedestrian. For example, both age and gender have a considerable influence on the pedestrians' decisions and, consequently, their future actions. The most influential attribute is the history of actions taken by the pedestrian, providing a clear description of the observed events. The tremendous evolution in human detection and action recognition models concedes the feasibility of obtaining these descriptive attributes. However, Herein, we rely on the ground truth existence of these features in our evaluation dataset.

6.3.4 Text Prompting

Much like human perception, where different synonyms of a single word can influence the comprehension of a sentence, language models are similarly sensitive to this phenomenon. Depending on the popularity of a given word and its contextual prevalence during the language model's training, the obtained representation would be more/less informative. Considering this effect, we leverage text prompting within captions to optimize the model's perception of the context and enhance its understanding of the described events.

6.3.4.1 Manual Prompting

For both ***"person_des"*** and ***"action_des"***, we prompt the captions by alternating their synonyms, producing multiple texts for the same image, allowing richer language representations. Table 6.2 summarizes the synonyms used for the prompting process. Given an initial template-based caption C^1_{temp}, a number P of prompted captions are created, where the collective representation of the projected captions $\mathbf{x}_{temp\_txt}$ is represented by (6.3).

$$\mathbf{x}_{temp\_txt} = Cat\Big[CLIP_{txt}(C^1_{temp}), \ldots, CLIP_{txt}(C^P_{temp})\Big] \tag{6.3}$$

6.3.4.2 ChatGPT Prompting

Recently, GPT models achieved a significant jump in the field of natural language processing, whereas ChatGPT showed impressive capabilities in multiple language tasks, including paraphrasing tasks. We asked ChatGPT to prompt our caption template into a set of possible paraphrased captions. To permit the production of more prompted captions through ChatGPT, the concatenation fusion approach in (6.3) is replaced with a summation.

6.3.4.3 Image Captions Prompting

The descriptions generated by an image captioner represent a broader interpretation of the images compared to a predefined captioning template. In contrast, template-based captioning provides a more to-the-point description of the event. To take advantage of both captioning techniques, image captions are prompted

Age Synonyms	Adult - Grownup Child - Little Child
Gender Synonyms	Female Male
Age/Gender Synonyms	Woman - Lady Man - Gentleman Girl - Schoolgirl Boy - Schoolboy
Action Synonyms	Walking - Moving Standing - Stopped - Was walking and stopped Waiting to cross - Wanting to cross Crossing the street - Crossing the road Does not want to cross Observed - Seen
ex. C^{1}_{temp}: An adult female is walking in the street C^{2}_{temp}: A woman is Walking in the street C^{3}_{temp}: A grownup female is moving in the street C^{4}_{temp}: A Lady is Walking in the street and does not want to cross	

Table 6.2: List of synonyms used in manual prompting

by our template, specifically, "***person_des***" and "***action_des***". Given an initial image caption C_{cap}, it is first prompted with the demographic age/gender attributes to generate a new caption C_{cap_D}. Then C_{cap_D} is prompted with the action attribute, generating C_{cap_AD}. The feature representation $\mathbf{x}_{cap\_txt}$ is provided in (6.4), where P is the number of template-promoting captions, S is the number of the considered spatial scales, while α and β are uncertainty values to reflect the possibility of erroneous descriptions generated by the captioner.

$$\mathbf{x}_{cap\_txt} = \sum_{s=1}^{S}(\alpha \times CLIP_{txt}(C^{s}_{cap}) + \beta \times CLIP_{txt}(C^{s}_{cap\_D}) + CLIP_{txt}(C^{s}_{cap\_AD})) + \sum_{p=1}^{P} CLIP_{txt}(C^{n}_{temp}) \tag{6.4}$$

6.4 Experimental Results

6.4.1 Dataset

We use a large-scale novel urban dataset LOKI [46] to evaluate our language-aided action anticipation approach (see Subsection 1.2.2). The dataset provides a larger-scale behavioral and demographical pedestrian annotation compared to older datasets. It comprises 9226 pedestrians, annotated, at 5 FPS, with age, gender, 2D bounding boxes, 3D bounding boxes, and intended action in the next 0.8 seconds (4 frames). The action labels for pedestrians consist of 5 actions: *Moving, Stopping, Waiting to cross, Crossing, and Other.* Due to the considerable imbalance in the dataset, we rely on the AUC and F1-score metrics for evaluation. Following [7, 117], we split the dataset into 60% training and 40% testing.

6.4.2 Implementation Details

Only the first training phase of TAMFORMER is considered to prove the effectiveness of the proposed approach. The model is trained for 500 epochs, using SGD optimizer and 10^{-3} learning rate. The earliest anticipation time is set to 4.0 s, and the warm-up window is 1 s. The time step α is constrained by the annotation rate of the dataset and set to 0.2 s. With respect to the bounding boxes, our spatial scaling ratios are represented by the range $[1.5\times, 5.0\times]$ with step $0.5\times$, in addition to the global scale, where $S = 9$ different scales. For manual text prompting, we set the number of prompts P to 4 prompted captions, while it is set to 10 for ChatGPT prompting. We employ three main modalities: Bounding boxes, the cropped image with a ratio $1.5\times$ of the bounding box, and the generated textual descriptions.

6.4.3 Results

As we are the first to work on the task of action anticipation on the LOKI dataset, we focus solely on the performance of our TAMFORMER model when upgraded to the non-binary anticipation of multiple actions, evaluating the influence of integrating language into the model.

Table 6.3 provides a comparative analysis of TAMFORMER's performance using various captioning techniques. As the anticipation space expands to more actions, the anticipation becomes increasingly complicated, resulting in poor performance.

	4 s		3 s		2 s		1 s	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
W/o Captioning	53.51	55.5	53.92	55.83	54.71	57.60	55.74	58.41
Predefined Captions	56.18	54.99	58.09	55.51	58.37	57.78	60.40	58.32
Image Captioner	56.34	57.65	57.96	58.18	61.28	57.97	63.98	60.39
Template Captions	70.10	64.51	70.77	65.14	72.98	70.10	80.80	77.68
Manual Prompting	77.30	71.05	80.86	75.95	84.64	80.92	89.67	87.40
ChatGPT Prompting	77.07	70.11	80.94	**76.24**	**85.35**	**81.34**	89.95	87.43
Captioner Prompting	**77.77**	**72.16**	**80.95**	75.98	85.24	**81.34**	**90.16**	**87.92**

Table 6.3: The performance comparison of TAMFORMER, with and without language integration, employing various captioning techniques, across different anticipation times $\{4.0, 3.0, 2.0, 1.0\}$ s.

Yet, the integration of any form of textual description into the model significantly improves anticipation performance. Notably, not all captioning approaches yield equally impactful enhancements. The predefined captioning approach, which selects the nearest predefined caption to the image in the $CLIP$ feature space, shows only marginal improvement over the model without textual input. This may be attributed to the possible limitation in added value compared to using image features alone, in addition to the restricted range of events and activities permitted in the predefined captions limiting the capabilities of the used captions. In contrast, employing a pre-trained image captioner allows for broader descriptions and interpretations, enriching the model with more valuable representations and yielding much-improved anticipations. However, our template-based captioning proves quite advantageous, offering more precise and accurate descriptions and achieving notably higher performance.

Textual prompting of all types shows impressive performance enhancement. Almost all prompting techniques are on par, yet the best performance is achieved by prompting the image captioner prompting, which takes advantage of both the generality of the pre-trained image captioner and the precise template-based captions. ChatGPT prompting allowed for increasing the number of prompted captions, richer representations, and high performance.

	4 *s*		3 *s*		2 *s*		1 *s*	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
1.5×	55.62	55.74	56.53	56.7	56.47	**58.36**	58.81	57.84
3.0×	54.61	56.64	**60.23**	54.33	56.75	57.96	59.40	58.28
5.0×	55.54	56.73	56.61	57.41	59.29	56.82	59.80	57.80
Global	55.43	56.62	59.70	54.82	58.18	56.3	59.40	57.13
Fusion	**56.34**	**57.65**	57.96	**58.18**	**61.28**	57.97	**63.98**	**60.39**

TABLE 6.4: The effect of changing the spatial scale of the cropped image in the generated captions and the performance.

6.4.4 Ablation Experiments

We conduct a set of ablation experiments to assess the effect of the different parameters in the captioning techniques utilized.

6.4.4.1 Spatial Scaling

Table 6.4 reports the effect of changing the spatial scale of the cropped image on the generated captions, leading to a shift in the anticipation performance. Each spatial scale in the image allows the captioner to capture a different perspective about the observed scene, as depicted in Figure 6.2. Therefore, combining different spatial scales widens the range of the captured information, offering better and more robust performance.

6.4.4.2 Manual Text Prompting

We test the effect of using different words describing the age/gender attributes on the anticipation performance, as shown in Table 6.5. In the first **T**ext **P**rompt (**TP**1), the word "person" is used in replacement of the age/gender attributes, *i.e.,* "*a person is walking in the street*", leading to degraded anticipation. Therefore, TP[2:6] show higher performance, as they employ the age/gender description. For each test, we fix the "***person_des***" to specific synonyms representing age and gender during training and evaluation. For example, TP2 uses "*adult/child*" for age and "*male/female*" for gender, while TP3 represents *age/gender* with "*woman/man/girl/boy*". Notably, alternating the words used in the captions affects the

	4 *s*		3 *s*		2 *s*		1 *s*	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
TP1	67.56	62.65	71.04	66.95	72.52	68.72	80.36	74.46
TP2	70.10	64.51	70.08	65.14	72.98	70.10	80.80	77.68
TP3	69.74	63.79	74.50	68.96	76.74	72.63	80.16	75.37
TP4	69.28	62.97	73.25	67.83	76.50	73.54	79.84	74.79
TP5	65.68	62.84	72.27	65.95	76.97	72.16	82.33	77.42
TP6	68.70	62.60	73.88	67.52	73.63	72.25	80.80	77.20
Fusion (TP2, TP3)	73.42	67.90	77.43	72.66	82.31	78.50	88.00	85.54
Fusion (TP[2:4])	76.35	70.58	80.25	74.37	83.59	79.17	**89.77**	**87.70**
Fusion (TP[2:4], TP$_{act}$)	**77.30**	**71.05**	**80.86**	**75.95**	**84.64**	**80.92**	89.67	87.40

Table 6.5: The effect of prompting the template-based captions with different synonyms representing the "***person_des***" and the "***action_des***". The **T**ext **P**rompts (**TP**[1:6]) prompts the "***person_des***", and **TP**$_{act}$ includes "***action_des***" prompting as well. The fusion technique used is as in Equation (6.3).

perception of the sentence in the language model, adding more/less information to the produced representation and influencing the performance.

Fusing multiple prompted captions improves the extracted information, compared to using a single caption, which leads to enhanced performance, as shown in Table 6.5. Additionally, as more captions are fused, the anticipation performance is better. Finally, adding TP$_{act}$, which prompts the "***action_des***", pushes the model to further performance improvement.

Table 6.6 reports the results of applying different fusion schemes on the prompted captions. As depicted, concatenating the text of captions and extracting the features of the concatenated texts, as in (6.5), shows a poor performance. Furthermore, increasing the number of concatenated captions decreases the performance even more. This degraded performance of the Cat_{txt} approach could be attributed to the increased complexity of the integrated text, making it harder for the CLIP model to provide a precise representation of the given text. Therefore, a finer approach is to apply the fusion in the feature space instead. Using the Avg_{feat} function, defined in (6.6), reports a better performance compared to the first approach, where the model is able to benefit from the integrated information. The best performance is achieved through the concatenation of features extracted from the integrated captions, as in (6.3).

	4 s		3 s		2 s		1 s	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
$Cat_{txt}(P=2)$	65.87	63.63	73.52	63.26	76.06	69.68	78.70	70.37
$Cat_{txt}(P=3)$	62.24	61.69	68.81	64.60	71.09	63.28	72.72	73.69
$Avg_{feat}(P=2)$	65.70	63.42	72.25	65.31	74.68	70.29	81.97	80.53
$Cat_{feat}(P=2)$	**73.42**	**67.90**	**77.43**	**72.66**	**82.31**	**78.50**	**88.00**	**85.54**

Table 6.6: Comparing different fusion approaches for the prompted captions. $Cat_{txt}(P)$, defined in Equation (6.5), denotes text concatenation, where the number of texts is P. $Avg_{feat}(P)$, defined in Equation (6.6), averages the extracted features from the captions, and $Cat_{feat}(P)$, defined in Equation (6.3), concatenate the extracted features.

$$Cat_{txt}(P) = CLIP_{txt}(Cat\left[C^{1}_{temp}, \ldots, C^{P}_{temp}\right]) \tag{6.5}$$

$$Avg_{feat}(P) = \frac{\sum_{p=1}^{P} CLIP_{txt}(C^{p}_{temp})}{P} \tag{6.6}$$

6.4.4.3 ChatGPT Prompting

Using ChatGPT for prompting introduces a broader range of prompts to the captions. In Table 6.7, we study the impact of increasing the number of prompts generated by ChatGPT. Firstly, comparing our template-based caption C_{temp} and a single randomly generated ChatGPT prompt demonstrates a clear preference for C_{temp}. ChatGPT tends to produce complex and general synonyms, such as "*strolling*" or "*wandering*" for the word "*walking*", potentially blurring the straightforward meaning of the caption and posing challenges for the feature extractor in providing informative representations. However, with increasing the number of prompts generated by ChatGPT, more informative descriptions are incorporated, resulting in improved performance. The peak performance is achieved with 10 generated prompts, beyond which the inclusion of additional prompts introduces numerous complex words, contributing to a decline in performance.

	4 s		3 s		2 s		1 s	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
C_{temp}	70.10	64.51	70.08	65.14	72.98	70.10	80.80	77.68
$GPT(P = 1)$	61.08	58.20	63.42	57.88	62.76	60.53	68.12	64.14
$GPT(P = 4)$	75.15	70.54	77.77	73.06	81.29	75.74	86.14	84.53
$GPT(P = 10)$	77.07	70.11	**80.94**	**76.24**	85.35	**81.34**	**89.95**	87.43
$GPT(P = 20)$	**77.02**	**71.29**	80.55	75.86	**85.54**	80.96	89.74	**87.93**

Table 6.7: Ablation on ChatGPT prompting, where P is the number of the prompted captions, GPT denotes ChatGPT prompting, and C_{temp} is the template caption on the form "*A /An* ***<adult /child> <male /female>*** *is* ***<movnig /stopping /waiting to cross /crossing /observed>***".

6.4.4.4 Image Captions Prompting

In this ablation study, we examine the impact of different levels of refining the image captions generated by a pre-trained captioner. The first two rows in Table 6.8 establish a comparison baseline using our template-based captioning approach. The first row represents the results of using a single templated caption, while the second row employs text prompting. The performance is notably subpar when directly using image captions without any refinement. A marginal improvement is observed when cleaning up the captions, as outlined in Section 6.3. The improvement, though limited, becomes more pronounced with the incorporation of multiple spatial scales, although it does not yet match the performance achieved with the template-based captioning approach. This observation can be attributed to the non-negligible noise present in the images, leading to captions with inherent noise. Consequently, refining the captions by incorporating accurate age, gender, and action attributes yields a remarkable enhancement in performance.

Comparing the refined image captions with the templated captions demonstrates the effectiveness of image captions in capturing additional information from the scene beyond the "***person_des***" and the "***action_des***" attributes used in our templates. The integration of the manual prompting procedure with refined image captions proves to be the most effective approach, surpassing the performance of all captioning techniques considered in our work.

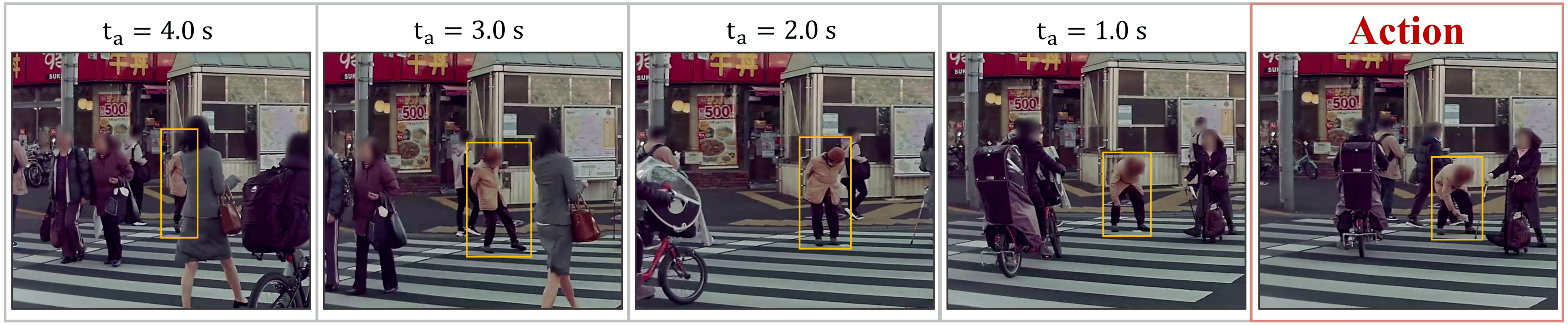
Cap	CU	MS	AR	MP	4 s		3 s		2 s		1 s	
					AUC	F1	AUC	F1	AUC	F1	AUC	F1
TC	-	-	-	✗	70.10	64.51	70.08	65.14	72.98	70.10	80.80	77.68
TC	-	-	-	✓	77.30	71.05	80.86	75.95	84.64	80.92	89.67	87.40
IC	✗	✗	✗	✗	55.62	55.74	56.53	56.7	56.47	58.36	58.81	57.84
IC	✓	✗	✗	✗	55.65	55.73	56.53	57.27	56.2	58.5	61.15	57.27
IC	✓	✓	✗	✗	56.34	57.65	57.96	58.18	61.28	57.97	63.98	60.39
IC	✓	✓	✓	✗	77.25	71.4	78.82	75.56	85.22	**81.93**	88.9	87.11
IC	✓	✓	✓	✓	**77.77**	**72.16**	**80.95**	**75.98**	**85.24**	81.34	**90.16**	**87.92**

TABLE 6.8: Studying the performance of the image captioning throughout the different levels of prompting and refinement. **TC** is the template-based captioning technique, and the **IC** reflects the usage of image captioner. For the image capions, we define the refinement levels as: **CU** denotes the cleanup process of the captions, **MS** represents the multi-spacial-scale fusion, and **AR** is the *age/gender/action* attributes refinement process, as described in Equation (6.4), but excluding the integration with the manually prompted captions $C_{temp}^{1:P}$. Finally, **MP** denotes the application of the manual prompts in either template-based captions or with the generated image captions.

6.4.5 Qualitative Example

Figure 6.4 presents a detailed case study on the generated image captions, focusing on a scenario where the pedestrian in question was initially crossing the street but was interrupted to pick something up and stopped. The template captions take advantage of the *age/gender/action* attributes at each frame, offering precise yet limited descriptions restricted to these attributes. In contrast, image captions provide broader descriptions but are vulnerable to generating errors or conveying misleading information. For instance, in Figure 6.4, due to the similarity between our pedestrian's body movements and those of someone engaged in skating, the captioner erroneously describes a skating activity in the images. Therefore, the incorporation of multi-scale information proves to be a necessity to capture a broader and more accurate understanding of the scene. The wider contexts in larger spatial scale captioning help the captioner align more accurately with urban scenarios activities.

Moreover, the generated captions have the potential to include more general yet valuable information, describing possible interactions within the scene, such as "A group of people walking down the street." In the refinement process, new captions are generated and integrated with the original ones, as detailed in Section 6.3.



	Template Caption	Image Caption (1.5 ×BBox) →	Refined Caption	Image Caption Higher Scale →	Refined Caption
$t_a = 4.0$s	an adult female is crossing the road	a woman in a tan top	a woman in a tan top	a woman is walking across a crosswalk	a woman is crossing across a crosswalk
$t_a = 3.0$s	an adult female is crossing the road	a group of people are walking down the street	a woman is crossing down the street	a woman is crossing the street	a woman is crossing the street
$t_a = 2.0$s	an adult female is crossing the road	a man is pushing a woman on a skateboard	a woman is crossing on a skateboard	a man is running across the street	a woman is crossing across the street
$t_a = 1.0$s	an adult female is stopped	a man is standing on a skateboard	a woman is stopped on a skateboard	a man is walking across the crosswalk	a woman is stopped across the crosswalk
Action	**Stopped**				

Figure 6.4: A qualitative example of image captioning and the refinement process.

These new captions aim to leverage the existing context in the original descriptions but are skewed more towards the accurate *age/gender/action* attribution. While the new captions might inherit some noisy contexts from the originals, they generally contribute to more accurate prompting in most cases, as evidenced by the enhanced performance outlined in Table 6.8.

6.5 Conclusion

In this chapter, we propose the integration of language descriptions in pedestrian action anticipation. We study multiple language generation approaches and tools, proving the effectiveness of coupling vision-language features to enrich the understanding of visual scenes and enhance anticipation performance. Additionally, we extended the binary crossing or not crossing pedestrian anticipation task into anticipating multiple actions, evaluating our task on a novel large-scale urban scenes dataset (LOKI). Our evaluation and ablation experiments demonstrate the outperformance of our language-aided anticipation approach, with an improvement over not using language, that reaches 29.5 % F1-score at 1-second anticipation and 16.66% at 4-second anticipation.

The generation of a textual description of an image is a fast-advancing field, where more informative and accurate captions could be generated, leading to even more improved performance, which is our focus in our future work.

Chapter 7

Traffic Flow Anticipation and Data Imputation

Chapter Abstract

This chapter addresses a specific type of behavior anticipation, anticipating traffic flow. Traffic flow represents the aggregate patterns of human behavior influenced by the road network and temporal variations throughout the day. We introduce a model for anticipating traffic speed, utilizing attention-based spatiotemporal encoding and a dual-graph road-network representation. The dual-graph framework combines spatial and contextual sub-graphs, facilitating the exploration of non-Euclidean spatial correlations and potential contextual similarities within road networks. To dynamically capture spatiotemporal correlation, we employ multi-head self-attention modules capable of discerning temporal and spatial correlations. Additionally, we extend to the problem of traffic data imputation, where we present a fast conditional diffusion model for spatiotemporal traffic data imputation, employing a high-order pseudo-numerical solver. [*]

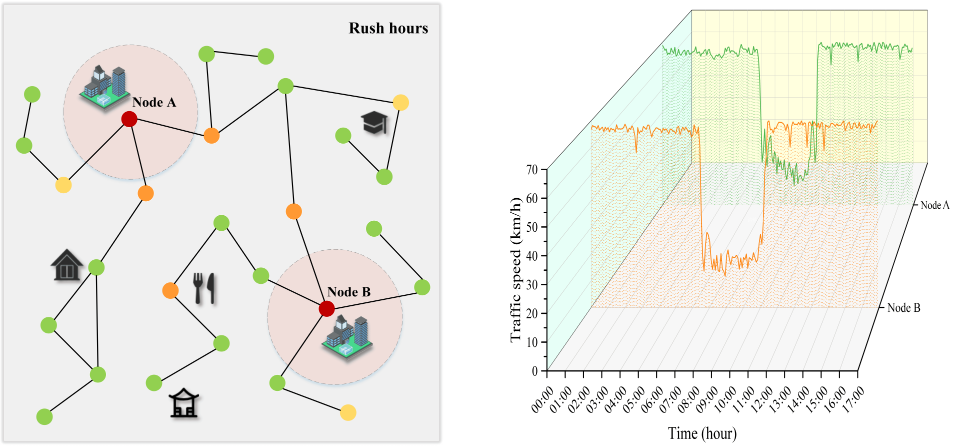
[*] This chapter has produced two submitted journal papers: 1) "DSP-ST: Dynamic Structural Prior Spatio-Temporal Graph Attention Networks for Traffic Speed Prediction" in *IEEE Intelligent Transportation Systems Magazine.* 2) "FastSTI: A Fast Conditional Pseudo Numerical Diffusion Model for Spatio-Temporal Data Imputation" in *IEEE Transactions on Intelligent Transportation Systems.*

7.1 Introduction

Traffic prediction is a foundational component of an Intelligent Transportation System (ITS) [83,120]. Accurate forecasting is crucial in improving traffic control, route optimization, and vehicle scheduling [79]. However, dynamic and accurate traffic forecasting is challenging due to the highly complex spatial and temporal correlations of roads (e.g., the coexistence of cyclicity, randomness, and fluctuating transmission) [121]. For decades, traffic prediction techniques have been extensively investigated. Early traditional methods are primarily statistical techniques [71, 72], which have visible limitations because of the nature of the assumption that time tends to be stationary. Classical ML methods [73, 74, 122] have been shown to outperform the statistical approaches on nonlinear and nonstationary traffic data. Still, these methods usually depend on human-engineered features, failing to capture complex spatial-temporal features for traffic prediction.

Deep learning (DL) approaches can approximate complex functions by learning deep nonlinear network structures to better mine the spatiotemporal evolution patterns of traffic conditions [86, 123]. The existing methods have extensively promoted the development of spatiotemporal traffic prediction, yet there are still open limitations:

- **The absence of implicit contextual information extraction**: similar traffic conditions in a physical road network may be implied between roads [82]. As depicted in Figure 7.1, two road nodes (Node A and Node B) located in a business district may experience similar traffic patterns during peak hours, with traffic congestion showing an upward or downward trend. However, most studies commonly employ the spatial adjacency matrix (e.g., distance-based matrix); as a result, the semantic information is ignored.
- **Lack of a dynamic capturing of long-term traffic features**: Existing time-series modeling methods, such as RNN and its variants LSTM and GRU, have received extensive attention in time-series analysis. However, the above RNN-based methods have limitations, such as time-consuming training, gradient explosion/vanishing, and slow response to dynamic changes. These methods fail to forecast long-term traffic accurately, where spatial traffic at different timestamps has a varying scale of impact on the target road node's pattern.



(A) Spatial location of similar nodes

(B) Comparison of traffic conditions of similar nodes

FIGURE 7.1: Traffic patterns between contextually similar nodes.

To address the abovementioned challenges, we propose Dynamic Structural Prior Spatio-temporal Graph Attention Networks (DSP-ST) to forecast traffic conditions. To extract the spatial correlation of roads more comprehensively, we designate a dynamic structural prior spatiotemporal graph architecture: the physical subgraph of road connectivity and the contextual subgraph constructed based on the tempo-feature similarity of different nodes. This graph is integrated into a spatio-temporal graph block with a self-attention mechanism to learn dynamically the spatio-temporal correlations. A Gated-TCN is used to extract the temporal correlation of long-range dependencies to improve the long-term predictions further.

Additionally, we address the problem of missing traffic data (traffic data imputation). Spatiotemporal traffic data is derived from diverse sensing systems such as loop detectors and floating cars, where equipment failures and transmission errors in these sensors leads to missing data and negatively affects the prediction task. Among various imputation methods, deep generative models have gained significant popularity. Recently, diffusion models have emerged as the new state-of-the-art method of deep generative models family [124–126], surpassing the long-standing dominance of generative adversarial networks (GANs) in diverse, challenging domains [127]. Existing works started to apply "Denoising Diffusion Probabilistic Model" (DDPM) to impute traffic missing data [128–130]. However, these approaches involve iterative procedures with several evaluation steps, which can be time-consuming and inefficient in real-time applications. Here, we design a fast conditional pseudo-numerical diffusion model for spatiotemporal traffic data imputation (FastSTI), where we apply pseudo-numerical methods and a predefined variance schedule to accelerate the inference time while retaining the imputation precision.

7.2 Problems Statement

The traffic network, a combination of intersections and roads, is a graphical structure. The road network can be denoted as a graph $G = (V, E, \boldsymbol{A})$, where V and E represent a finite set of vertices (intersections) and edges (roads), respectively. Let $N = |V|$ be the number of nodes, then the adjacency matrix $A \subseteq R^{N \times N}$ is a 0/1 matrix, representing the connections between each pair of nodes $(v_i,\ v_j)$. We define a dual-graph structure, i.e., a physical subgraph $G_P = (V, E_P, \boldsymbol{A_P})$, including the spatial information of the road nodes, and a contextual subgraph $G_C = (V, E_C, \boldsymbol{A_C})$, representing the contextual similarities between the nodes.

7.2.1 Flow Anticipation

Let $X = \{x_1, x_2, \ldots, x_L\}$ be a sequence of traffic flow signals (traffic speed), where $X \in R^{L \times N}$, L represents the length of time steps, and N is the number of observation nodes (e.g., observational points on the road map). The traffic signals at timestamp t are defined in the matrix $X_t \in R^{N \times 1}$, where N is the number of nodes. Traffic anticipation aims to learn a function $f(\ \cdot\ )$ that maps S_{obs} time steps of historical flow signals to future S_{ant} time steps of anticipated flows. Given our graphs G_P and G_C, the anticipation process at time t_a is represented by:

$$\Big[X_{(t_a - S_{obs})}, X_{(t_a - S_{obs}+1)}, \ldots, X_{(t_a - 1)}; G_P; G_C\Big] \overset{f(\ \cdot\ )}{\rightarrow} \Big[X_{(t_a)}, X_{(t_a+1)}, \ldots, X_{(t_a+S_{ant})}\Big] \tag{7.1}$$

7.2.2 Data Imputation

During data imputation, a binary mask is used to simulate the missing values, where the task is to estimate these values based on the observed traffic patterns. Our binary mask at imputation time step t_{imp} is given by $M_{t_{imp}} = \{0, 1\}^{N \times N}$, if $M^{i,j}_{t_{imp}} = 0$, the corresponding element $X^{i,j}_{t_{imp}}$ is missing, while $M^{i,j}_{t_{imp}} = 1$ denotes that $X^{i,j}_{t_{imp}}$ is an observed value. The objective of the task is to estimate the missing/masked signals in the data sequence.

7.3 Anticipation Model

Figure 7.2 illustrates the architecture of the proposed method. our DSP-ST is built upon the baseline model [86], which mainly stacks multiple spatio-temporal blocks (ST-blocks). In DSP-ST, each ST-block contains a self-attention Gated Temporal Convolution Network (Att-gated TCN) layer, along with the dual graph attention network (GAT) layers. Initially, feature mapping is applied with linear projection; then, the mapped features are concatenated into the ST-blocks. In the stacked ST-blocks, the temporal correlation of traffic conditions is dynamically extracted through the Att-gated TCN. Meanwhile, the spatial correlation is captured through the GAT module combining the dual graphs (i.e., physical and contextual sub-graphs). Finally, a linear anticipation head produces the predictions.

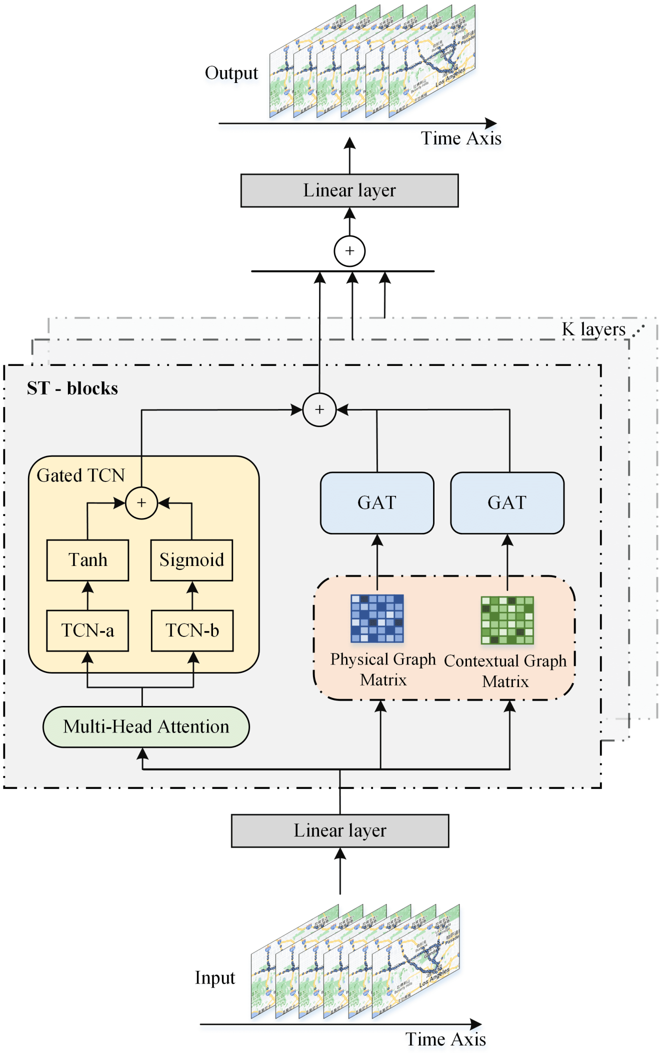


Figure 7.2: Overall structure of DSP-ST

7.3.1 Temporal Modeling

Traffic conditions present complex nonlinear changes in temporal evolution. To capture such nonlinear changes, a self-attention Gated-TCN is employed to extract the temporal correlation dynamically of traffic conditions. The Att-Gated-TCN module mainly comprises a self-attention module, dilated causal convolutions representing the temporal convolution, and gated activation units.

7.3.1.1 Multi-Head Attention

The temporal correlation of traffic characteristics is influenced by the overall interaction of the traffic network, plus a sort of randomness, as explained in [131]. To strengthen the model's capability to extract temporal features dynamically, we introduce a multi-head self-attention mechanism in the Gated-TCN. Given a time sequence input $X \in R^{N \times T}$, representing N nodes at T time steps, self-attention is computed over the input sequence. The attention is computed by $H_{att} = Att(X)$, producing a finer representation of the temporal correlations between the nodes, with the same size as X, where $H_{att} \in R^{N \times T}$.

7.3.1.2 Dilated Causal Convolution

Dilated causal convolution can exponentially improve the receptive field by increasing the layer depth and obtaining plenty of feature information [132]. Thus, following [86], we use dilated causal convolution as the temporal convolution layer. Given a 1D time sequence of attention representation $H^n \in R^T$ of length T at node n, and a filter $F(K)$ with kernel size K, dilated causal convolution takes the forms in (7.2), where t denotes the time step, and l represents the dilation factor.

$$H^n_{att}(t) *_l F(K) = \sum_{k=0}^{K-1} F(k) * H^n_{att}(t - l \times k) \tag{7.2}$$

7.3.1.3 Gated Activation

Following [86], the temporal features are extracted through a gated activation, applied after the TCN (Gated-TCN). For our hidden representation H_{att} of length T, the gated activation is illustrated in Equation (7.3), producing the final temporal representation H_{temp}, where σ is the sigmoid activation, $\odot$ is the element-wise product, and Θ_1 and Θ_2 are the kernels representing TCN-a and TCN-b respectively (see Figure 7.2).

$$H_{temp} = tanh(\Theta_1 *_l H_{att}) \odot \sigma(\Theta_2 *_l H_{att}) \tag{7.3}$$

7.3.2 Spatial Modeling

We employ a dual-graph representation of the traffic network: A physical graph to capture the physical spatial neighborhood of the nodes, and a contextual graph to capture the semantic correlations between the nodes.

7.3.2.1 Physical Graph Construction

The physical connectivity of roads is one of the critical factors to consider in spatial correlation, where a pair of road nodes (v_i and v_j) have a physical/spatial neighbor relationship when they share the same intersection. Hence, the physical graph $G_P = (V, E_P, \boldsymbol{A_P})$ defines the spatial adjacency matrix A_P as follows:

$$[\boldsymbol{A}_p]_{ij} = \begin{cases} 1, & v_i \text{ and } v_j \text{ are directly connected} \\ 0, & \text{otherwise} \end{cases} \tag{7.4}$$

Intuitively, points nearby in a road network are more likely to share similar traffic patterns. In [48, 85, 133], the Euclidean distances between nodes represent the spatial neighbor relationship as edge weights, where the weight increases as the distance decreases.

7.3.2.2 Contextual Graph Construction

The traffic network always has contextual similarities between roads at different times. For example, in Figure 7.1, the nodes A and B do not have a physical connection yet show a similar traffic pattern at a specific time period. Therefore, to capture such contextual information of the traffic network, we build a contextual subgraph $G_C = (V, E_C, \boldsymbol{A_C})$, adopting the improved Derivative Dynamic Time Warping (DDTW) [134] to calculate the similarity between the traffic time series.

Dynamic Time Warping (DTW) is widely used for finding an optimal alignment between two-time series under certain conditions [135]. However, the traditional

DTW calculates the similarity by matching the numerical difference between the two sequences but tends to ignore the trend information. In Figure 7.3a, the two nodes A and B show similar traffic sequences; using the traditional DTW, as in Figure 7.3b, a mismatching occurs between 9:00 and 10:30, where the downtrend information of node A matched the uptrend information of node B. In other words, the downtrend and uptrend of traffic sequence may correspond to the beginning and end of road congestion, where the traditional DTW cannot accurately reflect the similar trend information between the two sequences.

To overcome the limitation that the traditional DTW algorithm fails to study the trend information, we use the derivatives to preprocess the initial data to reflect the shape characteristics and the trends of the values. The DDTW matching strategy is shown in Fig. 7.3c, which can reflect the difference in values and consider the trend information between the two sequences. Given two initial time sequences $A = (a_1, a_2, ..., a_{T_A})$ and $B = (b_1, b_2, ..., b_{T_B})$, where T_A and T_B represent the lengths of the sequences. Firstly, the elements of sequences are preprocessed to estimate the derivatives, as in (7.5), and the Z-Score normalization, as in (7.6).

$$A' = \begin{cases} a_i' = \frac{(a_i - a_{i-1}) + ((a_{i+1} - a_{i-1})/2)}{2} & 2 \leq i < T_A \\ a_i' = a_2 - a_1 & i = 1 \\ a_i' = a_{T_A} - a_{T_A - 1} & i = T_A \end{cases} \tag{7.5}$$

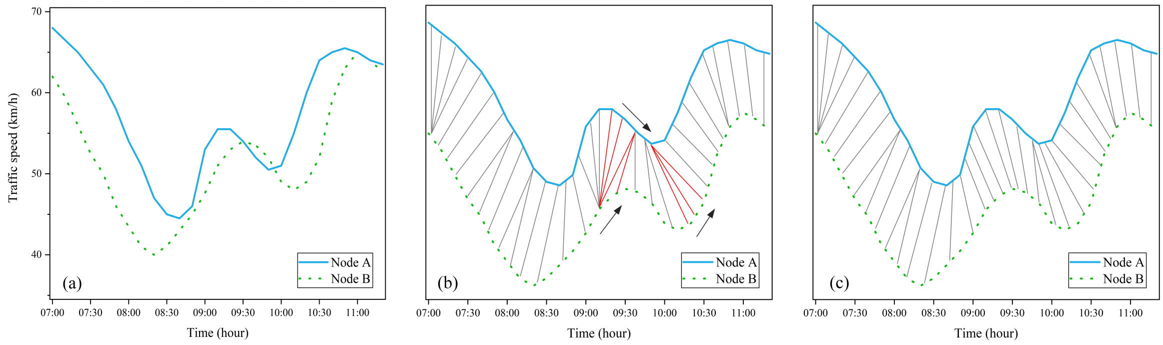
Where $a_i^{'}$ is the i^{th} element of the sequence, and μ_A and σ_A denote the mean and standard deviation of A, respectively

$$a_{i^*}^{'} = \frac{a_i^{'} - \mu_A}{\sigma_A} \tag{7.6}$$

We can compute the Euclidean distance between $a_{i^*}^{'}$ in A and $b_{j^*}^{'}$ in B as in (7.7). To reduce the complexity of DDTW, we restrict its "Search Length" to T_{search}.

$$d(a_{i^*}^{'}, b_{j^*}^{'}) = \sqrt{(a_{i^*}^{'} - b_{j^*}^{'})^2}, i, j < T_{search} \tag{7.7}$$

Let l be a random alignment between the elements in A and the elements in B, and L is the number of all possible alignments between the two sets. The DDTW distance d_{DDTW} between A and $B$is given by (7.8)



(A) The original sequences

(B) Traditional DTW matching

(C) The modified DDTW matching

Figure 7.3: Traffic speed similarity matching between two sequences.

$$d_{DDTW}(A,B) = \underset{l\in L}{min} \sum_{(i,j)\in l} d(a^{'}_{i*}, b^{'}_{j*}) \tag{7.8}$$

Accordingly, we define the contextual subgraph $G_C = (V, E_C, \boldsymbol{A_C})$ through the DDTW distance of the traffic speed among road nodes. The contextual adjacency matrix A_C is defined in (7.9), representing whether the nodes are contextually neighbors, where τ denotes a threshold to control the sparsity of the matrix.

$$[A_c]_{ij} = \begin{cases} 1, & d_{DDTW} < \tau \\ 0, & \text{otherwise} \end{cases} \tag{7.9}$$

7.4 Imputation Model

Figure 7.4 illustrates the pipeline of the proposed FastSTI. FastSTI is built on top of the state-of-the-art PriSTI model [130]. In FastSTI, the input $X \in R^{N\times T}$ of complete observation is masked to simulate the missing data. Random sampling, representing noise, and linear interpolation, representing conditional prior, are applied to the masked values. The conditional features prior module is used to project the prior features, and then the projected prior and the noisy samples are fed into the noise prediction module, producing the imitated values.

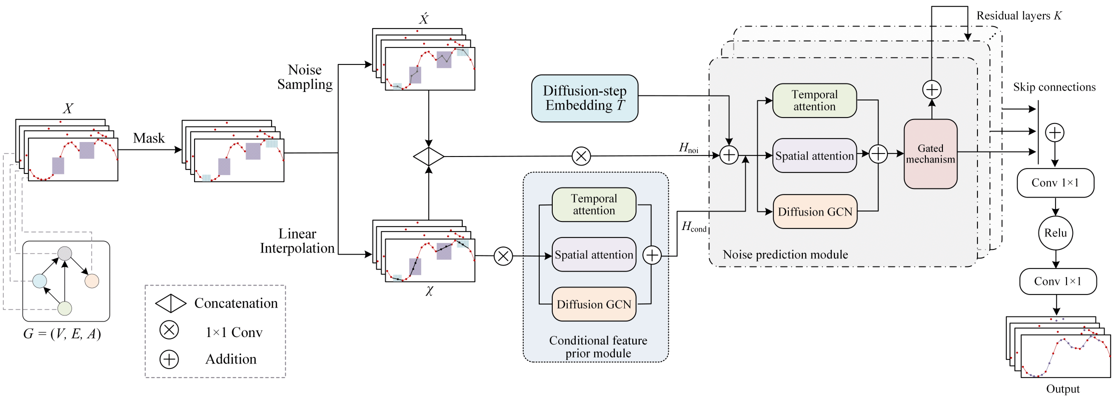


FIGURE 7.4: FastSTI pipeline.

7.4.1 Masking Strategy

Given the observed sequence X, we split it into two parts: one represents the imputation target $\acute{X}$, while the other represents the observed values serving as conditional observations. To simulate different real-life scenarios of missing traffic data, following [136], we consider two masking strategies:

- **Block-missing scenario:** Missing values occur in contiguous blocks over time. We randomly mask 5% of the available data and adopt simulated failures with a probability of 0.15% for each node/sensor. The duration of each failure is sampled uniformly from the interval $[min\_steps, max\_steps]$, where min_steps and max_steps correspond to the length of time steps.
- **Point-missing scenario:** Random occurrence of missing values, where 25% of observations are masked in a random manner.

7.4.2 Linear Interpolation

Following [130], linear interpolation is employed to fill in the missing data at each node. This approach can construct a rough but efficient interpolated conditional information χ for denoising. The linear interpolation relies on the uniform distribution of missing data to ignore the randomness of time series but retains certain spatiotemporal relations.

7.4.3 Conditional Feature Prior Module

The linear interpolation method assumes uniform and linear changes in traffic states. However, traffic data exhibits dynamic temporal dependencies, and the flow of different regions/interactions affects each other, making linear interpolation inadequate for capturing the nonlinear and random patterns in real-life traffic conditions [137]. Therefore, [130] employed a learnable conditional prior module $\rho(\,\cdot\,)$. It takes the interpolated information χ and the adjacency matrix $\mathcal{A}$ to model the nonlinear conditional information H_{cond} that represents the global spatiotemporal and local geographic correlations, as given in (7.12), where $\mathcal{H}$ is a 1×1 convolution of the interpolated data ($\mathcal{H} = Conv(\chi)$). To capture the global temporal correlation, an attention module $\phi_{Tem}(\mathcal{H})$ is applied in (7.13), while another global attention module $\phi_{Spa}(\mathcal{H})$ captures the global spatial correlations in (7.14).

PriSTI [130] uses a simple message-passing network (MP) to capture the spatial geographic correlation, where $\phi_{\text{SGC}}(\mathcal{H}, \mathcal{A})) = \phi_{\text{MP}}(\mathcal{H}, \mathcal{A}))$. Unlike PriSTI, we model the spatial correlation using a graph convolution network module named Diffusion-GCN (Diff-GCN) [48], defined in (7.10) and (7.11), where $\varrho \in [0,1]$ is the graph coefficient, and k is the graph convolution step. In contrast to the MP, the Diff-GCN benefits from a bidirectional random walk strategy, providing enhanced flexibility in capturing influences from upstream and downstream traffic conditions.

$$\text{DiffGCN}(\mathcal{H}, \mathcal{A}) = \varrho \sum_{k=0}^{K} A\chi W \tag{7.10}$$

$$\phi_{\text{SGC}}(\mathcal{H}, \mathcal{A}) = \text{Norm}\left(\text{DiffGCN}(\mathcal{H}, \mathcal{A}) + \mathcal{H}\right) \tag{7.11}$$

$$\rho(\mathcal{H}, \mathcal{A}) = \text{MLP}(\phi_{Tem}(\mathcal{H}) + \phi_{Spa}(\mathcal{H}) + \phi_{\text{SGC}}(\mathcal{H}, \mathcal{A})) \tag{7.12}$$

$$\phi_{Tem}(\mathcal{H}) = \text{Norm}\left(Attn_{Tem}(\mathcal{H}) + \mathcal{H}\right) \tag{7.13}$$

$$\phi_{Spa}(\mathcal{H}) = \text{Norm}\left(Attn_{Spa}(\mathcal{H}) + \mathcal{H}\right) \tag{7.14}$$

7.4.4 Noise Prediction Module

The noise prediction module is designed to utilize conditional information to predict the missing imputation values, as shown in Fig. 7.4. The module takes two inputs: 1) The conditional prior H_{cond}; 2) Noise information $H_{noi} = Conv(\chi || \acute{X})$, where $Conv(.)$ is the 1×1 convolution, $(||)$ represents concatenation, and $\acute{X}$ is the data sequence with the missing data sampled from a standard Gaussian noise. The temporal, spatial, and geographic modules of the noise prediction are the same modules of the conditional feature prior. However, the noise information H_{noi} would not provide an accurate representation of real-life traffic data; therefore, the conditional information H_{cond} is used for both the query and value of the temporal attention ϕ_{Tem} and spatial attention ϕ_{Spa}, while H_{noi} is used for the key.

The output of each layer in the noise prediction module is split into a residual connection and skip connections. The residual connection is the input of the next layer, and the skip connections of each layer are added and fed into a two-layer 1×1 convolution to obtain the output of the noise prediction module, where the output contains only the value of the imputation target.

7.4.5 Diffusion Procedure

Following [130], given the observed sequence X and split into: The imputation target $\acute{X}$, generated by the masking strategies, in addition to the remaining observed values (conditional observations) χ. The noise predictor ϵ_θ, is trained to minimize the loss function $\mathcal{L}_t$, defined in (7.15), where the imputation target $\acute{X}_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, β_t is the noise level in the diffusion model, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and ϵ is the sampled Gaussian noise.

$$\min\mathcal{L}_t = \min\mathbb{E}_{\acute{X}\sim q(\acute{X}_0),\epsilon\sim N(0,I)} \,||\, \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\acute{X}_t, \chi, A, t) \,||^2 \tag{7.15}$$

7.4.5.1 Pseudo-Numerical Diffusion

We aim to speed up the imputation process and improve its quality. PriSTI [130], and CSDI [128] directly employ the reverse process of the denoising diffusion probabilistic model (DDPM), which requires a sufficient number of denoising steps and encounters random noise. To tackle the issue, we apply the high-order pseudo-numerical methods [138] for denoising. The adopted methodology starts with transforming the reverse process of diffusion, given in (7.16), into an ordinary differential equation by subtracting x_t from both sides, replacing the discrete $t-1$ with a continuous version represented by $t - \Delta t$ and allowing *Deltat* to reach 0, producing the differential equation given in (7.17).

$$x_{t-1} = \sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\theta(x_t, t) \right) + \sqrt{1-\bar{\alpha}_{t-1}} \epsilon_\theta(x_t, t) \tag{7.16}$$

$$\frac{dx}{dt} = -\bar{\alpha}'(t) \left(\frac{x(t)}{2\bar{\alpha}(t)} - \frac{\epsilon_\theta(x(t), t)}{2\bar{\alpha}(t)\sqrt{1-\bar{\alpha}(t)}} \right) \tag{7.17}$$

The pseudo-numerical method applies three classical numerical techniques: Heun's, the Runge-Kutta (RK), and the Linear Multi-Step (LMS). The numerical methods are split into two components: a gradient component responsible for determining the gradient at each step and a transfer component generating the result for the next step. All numerical methods share the same transfer part, given in (7.18), while their gradient parts differ, as given in Table 7.1.

$$\begin{aligned} &\nu\left(x_t, \epsilon_t, \chi, A, t, t-\Delta t\right) = \\ &\frac{\sqrt{\bar{\alpha}_{t-\Delta t}}}{\sqrt{\bar{\alpha}_t}} x_t - \frac{\left(\bar{\alpha}_{t-\Delta t} - \bar{\alpha}_t\right)}{\sqrt{\bar{\alpha}_t}\left(\sqrt{\left(1-\bar{\alpha}_{t-\Delta t}\right)\bar{\alpha}_t} + \sqrt{\left(1-\bar{\alpha}_t\right)\bar{\alpha}_{t-\Delta t}}\right)} \epsilon_t \end{aligned} \tag{7.18}$$

We provide two kinds of pseudo-numerical methods for the conditional diffusion model in our task: **FastSTI-2** (2^{nd}-order) and **FastSTI-4** (4^{th}-order), where we initially adopt the 4th-order pseudo Runge-Kutta (PRK4) method to obtain the results of the first three steps, followed by the utilization of the 4^{th}-order pseudo-linear multi-step method (PLMS4) to compute the remaining.

7.4.5.2 Accelerated Diffusion

To accelerate the imputation (sampling) process, inspired by [139], we introduce a "schedule alignment" approach that utilizes a predefined number of T_{acc}-steps to minimize the imputation time without significant loss of quality. As shown in Fig. 7.5, the key concept is to align the original T-steps reverse process into a condensed T_{acc}-steps process using a predefined variance schedule. Given the steps of $T_{acc} \ll T$ in the imputation process and a predefined variance schedule $\{\xi_t\}_{t=1}^{T_{acc}}$, we can calculate the corresponding constants as in (7.19). The objective is to determine the value of $\sqrt{\bar{\varphi}_c}$ between the training noise levels: $\sqrt{\bar{\alpha}_t}$ and $\sqrt{\bar{\alpha}_{t+1}}$, such that $\sqrt{\bar{\varphi}_c}$ closely approximates $\sqrt{\bar{\alpha}_t}$. Then, the aligned diffusion step t is obtained by calculating the floating-point t_c, as in (7.20).

2^{nd}-order pseudo linear multi-step (PLMS2)
$\begin{cases} e_t = \epsilon_\theta\left(\acute{x}_t, \chi, A, t\right), \\ e'_t = \frac{1}{2}(3e_t - e_{t-\Delta t}), \\ x_{t-\Delta t} = \nu(\acute{x}_t, \chi, e_t^{'}, A, t, t-\Delta t). \end{cases}$
2^{nd}-order pseudo Heun's (PH2)
$\begin{cases} e_t^1 = \epsilon_\theta\left(\acute{x}_t, \chi, A, t\right), \\ x_t^1 = \nu(\acute{x}_t, \chi, e_t^1, A, t, t-\Delta t), \\ e_t^2 = \epsilon_\theta\left(\acute{x}_t^1, \chi, A, t-\Delta t\right), \\ e'_t = \frac{1}{2}(e_t^1 + e_t^2), \\ x_{t-\Delta t} = \nu(\acute{x}_t, \chi, e_t^{'}, A, t, t-\Delta t). \end{cases}$
4^{th}-order pseudo linear multi-step (PLMS4)
$\begin{cases} e_t = \epsilon_\theta\left(\acute{x}_t, \chi, A, t\right), \\ e'_t = \frac{1}{24}(55e_t - 59e_{t-\Delta t} + 37e_{t-2\Delta t} - 9e_{t-3\Delta t}), \\ x_{t-\Delta t} = \nu(\acute{x}_t, \chi, e_t^{'}, A, t, t-\Delta t). \end{cases}$
4^{th}-order pseudo Runge-Kutta (PRK4)
$\begin{cases} e_t^1 = \epsilon_\theta\left(\acute{x}_t, \chi, A, t\right), \\ x_t^1 = \nu(\acute{x}_t, \chi, e_t^1, A, t, t-\frac{\Delta t}{2}), \\ e_t^2 = \epsilon_\theta(\acute{x}_t^1, \chi, A, t-\frac{\Delta t}{2}), \\ x_t^2 = \nu(\acute{x}_t, \chi, e_t^2, A, t, t-\frac{\Delta t}{2}), \\ e_t^3 = \epsilon_\theta(\acute{x}_t^2, \chi, A, t-\frac{\Delta t}{2}), \\ x_t^3 = \nu(\acute{x}_t, \chi, e_t^3, A, t, t-\Delta t), \\ e_t^4 = \epsilon_\theta\left(\acute{x}_t^3, \chi, A, t-\Delta t\right), \\ e'_t = \frac{1}{6}(e_t^1 + 2e_t^2 + 2e_t^3 + e_t^4), \\ x_{t-\Delta t} = \nu(\acute{x}_t, \chi, e_t^{'}, A, t, t-\Delta t). \end{cases}$

TABLE 7.1: Gradient equations of the different pseudo numerical methods [138]

$$\varphi_t = 1 - \xi_t,\ \bar{\varphi}_t = \prod_{s=1}^{t} \varphi_c,\ \tilde{\xi}_t = \frac{1-\bar{\varphi}_{t-1}}{1-\bar{\varphi}_t}\xi_t \tag{7.19}$$

$$t_c = t + \frac{\sqrt{\bar{\alpha}_t} - \sqrt{\bar{\varphi}_c}}{\sqrt{\bar{\alpha}_t} - \sqrt{\bar{\alpha}_{t+1}}} \tag{7.20}$$

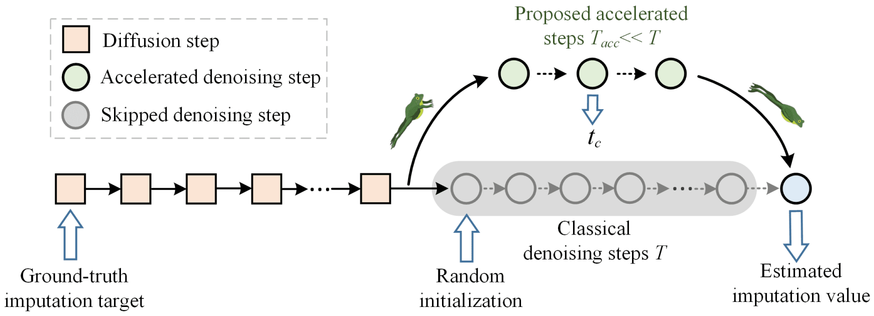

Diffusion step
Accelerated denoising step
Skipped denoising step
Proposed accelerated steps $T_{acc} << T$
t_c
Classical denoising steps T
Ground-truth imputation target
Random initialization
Estimated imputation value

FIGURE 7.5: Accelerated imputation. FastSTI utilizes "schedule alignment" to estimate the denoised distribution, replacing multiple classical denoising steps, thereby accelerating inference without significant loss of accuracy.

7.5 Experimental Results

7.5.1 Datasets

As detailed in Subsection 1.2.3, we conducted our experiments on two publicly available real-life traffic datasets: METR-LA and PEMS-BAY [48]. METR-LA comprises four months of traffic speed statistics collected from 207 loop detectors installed on the highway of Los Angeles County. Similarly, PEMS-BAY holds six months of traffic speed data from 325 sensors in the Bay Area.

7.5.2 Evaluation Metrics

7.5.2.1 Anticipation Metrics

We followed the same evaluation protocol as in [135, 140] to provide a fair comparison with the state-of-the-art. The data sets are split into 70% for training, 10% for validation, and 20% for testing. The model takes the historical traffic speed of one hour and predicts the expected traffic speed for the next 15, 30, or 60 minutes. Three evaluation metrics are used: Mean Absolute Error (MAE), as in (7.21); Root Mean Square Error (RMSE), described in (7.22); and Mean Absolute Percentage Error (MAPE), described in (7.23).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \acute{y}_i| \tag{7.21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \acute{y}_i)^2} \tag{7.22}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \acute{y}_i}{y_i}| \tag{7.23}$$

7.5.2.2 Imputation Metrics

For imputation, we followed the evaluation protocol in [128, 130]. Again, the data sets are split into 70% for training, 10% for validation, and 20% for testing. Three evaluation metrics are adopted: Mean Absolute Error (MAE), as in (7.21); Mean Square Error (MSE), described in (7.24); and Continuous Ranked Probability Score (CRPS) [141]. CRPS reflects the compatibility between a missing value x and its estimated probability distribution D. CRPS is calculated as the integral of the quantile loss Λ_ω, as in (7.25), where ω is the quantile level and $D^{-1}(\omega)$ is the ω-quantile of the distribution D. The quantile loss is defined in (7.26).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}((x_i - \acute{x}_i) \odot m_{eval})^2 \tag{7.24}$$

$$\text{CRPS}(P^{-1}, x) = \int_0^1 2\Lambda_\omega(P^{-1}(\omega), x)d\omega \tag{7.25}$$

$$\Lambda_\omega(D^{-1}(\omega), x) = (\omega - 1_{x<D^{-1}(\omega)})(x - D^{-1}(\omega)) \tag{7.26}$$

Following the same setting in [128], 100 samples are generated to approximate the distribution of missing values. We compute quantile losses for discretized quantile levels with 0.05 ticks in Equation (7.27). The final metric is the average CRPS computed at of the missing points $\text{CRPS}(D, \widetilde{X})$, as in (7.28).

$$\text{CRPS}(D^{-1}, x) \simeq \sum_{i=1}^{19} 2\Lambda_{i*0.05}(D^{-1}(i*0.05), x)/19 \tag{7.27}$$

$$\text{CRPS}(D, \widetilde{X}) = \frac{\sum_{\widetilde{x}\in\widetilde{X}} \text{CRPS}(D^{-1}, \widetilde{x})}{\left|\widetilde{X}\right|} \tag{7.28}$$

Hyperparameter	Value
Epochs	200
Batch size	16
Sequence length L	24
Learning rate	1×10^{-3}
Weight decay	1×10^{-6}
Residual layers	4
Residual channels r	64
Self-attention heads h	8
Temporal embedding dim m	128
Diffusion Schedule	Quadratic
The minimum noise level β_1	01
The maximum noise level β_T	0.2
Diffusion steps T	50
Accelerated denoising steps T_{acc}	6
Variance Schedule	$\{01, 1, 0.2, 0.3, 0.5, 0.9\}$

Table 7.2: FastSTI hyperparameters

7.5.3 Implementation Details

7.5.3.1 Anticipation Model

We train the model for 100 epochs, where the batch size is 64, using the Adam optimizer with a learning rate of 1 and setting the dropout value to 0.2. To extract the contextual graph, we set the search length in the DDTW algorithm to $T_{search} = 12$, the longest prediction time step in the DSP-ST model. We tuned the hyper-parameters of the model to the following values: The threshold of controlling the sparsity of the contextual matrix is set to $\tau = 0.85$, the number of attention heads is 3, the hidden dimension of GAT is 32, and the number of the stacked ST-blocks is $S = 2$.

7.5.3.2 Imputation Model

Table 7.2 summarizes our hyperparameters, where the model is trained for 200 epochs with a batch size of 16, learning rate of 10^{-3}, and Adam optimizer. The diffusion model noise schedule is Quadratic, and we utilize user-defined variance schedules $\{01, 1, 0.2, 0.3, 0.5, 0.9\}$.

7.5.4 Anticipation Results

We compare the DSP-ST model with nine baseline models, including statistical, classical ML, and mainstream DL models in the traffic prediction domains, as provided in Table 7.3. Our model DSP-ST outperforms other baseline models on PEMS-BAY and METR-LA datasets. Traditional traffic prediction methods such as ARIMA [72] and SVR [142] have the highest prediction errors. These methodologies cannot handle complex nonlinear traffic data, relying on historical time series features and ignoring the spatial correlation. For the DL-based models, the STGCN [143] and DCRNN [48] can simultaneously capture spatial and temporal correlations. Still, these methods only use a predefined spatial/physical adjacency matrix to construct a graph convolution network for spatial correlation modeling but overlook semantic neighbor features. The Graph WaveNet [80] utilizes the self-learning adjacency matrix to extract spatial features; however, it has a weak ability to extract temporal features dynamically. Conversely, PGCN [86] combines multiple graphs, such as the structural graph, self-adaptive graph, and progressive graph, to capture the spatial correlation of traffic networks. However, the focus of these graphs is mainly on local information aggregation and updates. In contrast, Our DSP-ST considers both the local physical sub-graph and the global contextual sub-graph, thereby enabling more effective learning of spatial correlations.

AA visualization of the forecasting curves of one hour on the PEMS-BAY and METR-LA datasets is given in Figure 7.6. The DSP-ST model has a satisfactory performance in data fitting for both datasets, proving that, regardless of the curve smoothness, the DSP-ST model has a promising ability to learn the traffic conditions and anticipate future flows.

7.5.5 Anticipation Ablation Experiments

We first analyze the effect of the sparsity hyperparameter τ on the RMSE performance using the METR-LA dataset. As seen in Table 7.4 and Figure 7.7, there is a moderate RMSE performance and computational cost when the threshold value range is [0,1]. To balance both performance and computation cost, we chose 0.85 as the optimal value.

We conduct another ablation study on the METR-LA datasets to verify the impact of the different components of our DSP-ST model: **GAT w/o**, DSP-ST without the GAT module, replaced by a GCN module; **CG w/o**, DSP-ST without the contextual graph; **Multi-Att w/o**; DSP-ST without the multi-head attention

Datasets	Models	15min (t=3)			30min (t=6)			60min (t=12)		
		e_1	e_2	e_3	e_1	e_2	e_3	e_1	e_2	e_3
METR-LA	ARIMA [72]	3.99	9.60	8.21	5.15	12.70	10.45	6.90	17.40	13.23
	SVR [142]	3.76	8.34	7.63	4.58	11.57	9.24	6.01	15.28	12.19
	STGCN [143]	2.88	7.62	5.74	3.47	9.57	7.24	4.59	12.70	9.40
	DCRNN [48]	2.77	7.30	5.38	3.15	8.80	6.45	3.60	10.50	7.60
	Graph WaveNet [80]	.69	6.90	.15	3.07	8.37	6.22	3.53	10.01	7.37
	GMAN [144]	2.75	7.19	5.42	.01	.18	6.25	.34	.71	.21
	MTGNN* [81]	.69	.89	5.22	3.07	8.19	.19	3.52	9.84	7.23
	GTS [145]	**2.67**	7.21	5.27	3.04	8.41	6.25	3.46	9.98	7.30
	PGCN [86]	2.70	6.98	5.16	3.08	8.38	6.22	3.54	9.94	7.36
	DSP-ST (ours)	2.71	**6.71**	**5.06**	**2.94**	**7.85**	**6.11**	**3.31**	**9.67**	**7.15**
PEMS-BAY	ARIMA [72]	1.62	3.50	3.30	2.33	5.40	4.76	3.38	8.30	6.50
	SVR [142]	1.58	3.41	3.43	2.29	4.98	4.53	3.01	7.25	6.43
	STGCN [143]	1.36	2.90	2.96	1.81	4.17	4.27	2.49	5.79	5.79
	DCRNN [48]	1.38	2.90	2.95	1.74	3.90	3.97	2.07	4.74	4.90
	Graph WaveNet [80]	.30	2.73	.74	1.63	3.67	.70	1.95	4.52	4.63
	GMAN [144]	1.34	2.81	2.82	.62	.63	3.72	.86	.31	.32
	MTGNN* [81]	1.35	2.75	2.83	1.65	3.68	3.76	1.89	4.50	4.49
	GTS [145]	1.34	2.82	2.83	1.66	3.78	3.77	1.95	4.58	4.43
	PGCN [86]	.30	**2.72**	.74	.62	.63	**3.67**	1.92	4.55	4.45
	DSP-ST (ours)	**1.28**	.75	**2.65**	**1.58**	**3.60**	**3.67**	**1.85**	**4.24**	**4.18**

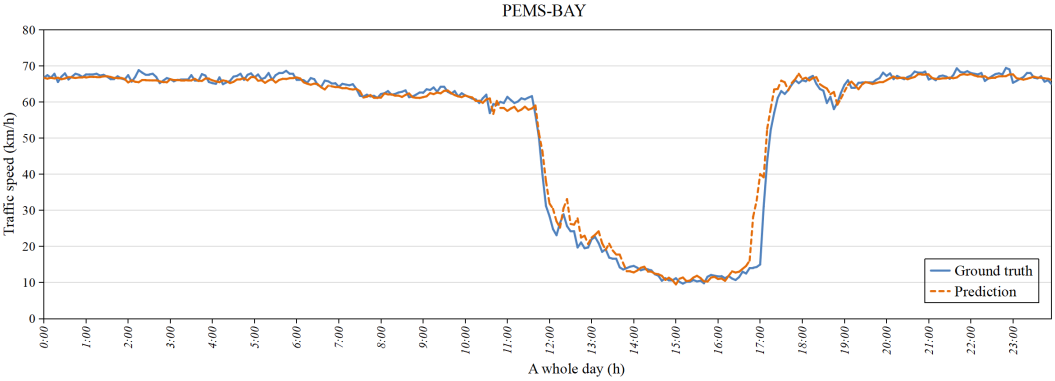
Table 7.3: Performance comparison of DSP-ST and other baseline models (bold = best, underline = second best). MTGNN* denotes our retrain of the model. e_1 is the MAE, e_2 is the MAPE (%), and e_3 is the RMSE.

τ	Training time (s/epoch)	Inference time (s)
0.6	28.55	8.71
0.8	16.48	3.26
0.85	15.33	3.11
0.9	14.42	2.41
0.95	13.80	2.25

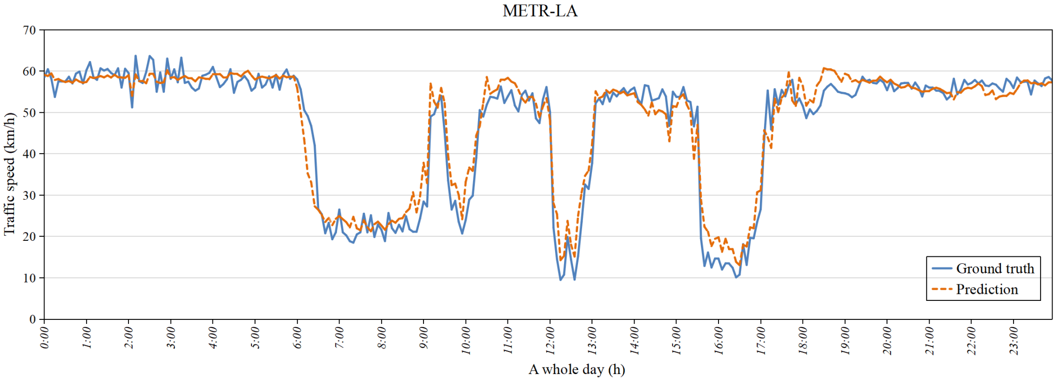
Table 7.4: The computational cost the sparsity threshold τ

for the temporal correlation modeling; **Gating w/o**; DSP-ST without the gating mechanism at the TCN.

Figure 7.8 shows the average scores of MAE and RMSE over one-hour prediction, compared to the complete DSP-ST model. There is a notable drop in the prediction performance, especially for long-term forecasting (i.e., > 30 min) when GCN replaces the GAT module (GAT w/o), where the edges among nodes are static



(A) PEMS-BAY



(B) METR-LA

FIGURE 7.6: DSP-ST prediction curves on PEMS-BAY and METR-LA datasets

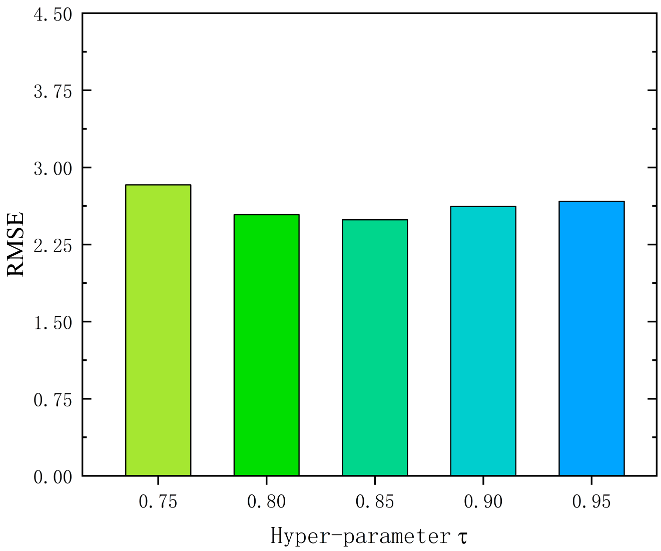


FIGURE 7.7: The RMSE performance of the sparsity threshold τ

weights rather than dynamic, resulting in a weak ability to extract spatial dependencies dynamically. The same is observed with the removal of the multi-head attention module (Multi-Att w/o). When the contextual graph is excluded (CG w/o), the model no longer considers the semantic similarity information among

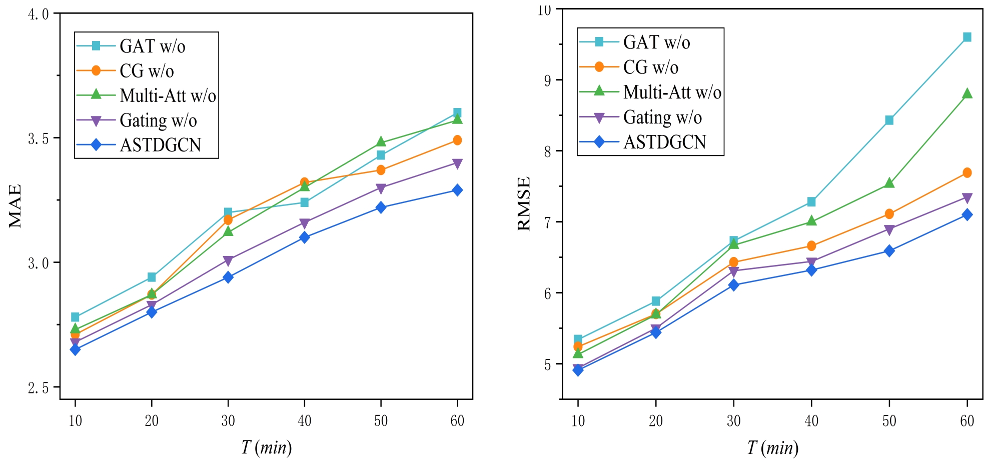


Figure 7.8: Ablation study on DSP-ST

nodes on the traffic road network, leading to a decrease in the prediction accuracy, as there is no extraction of the deep latent spatial correlation between the nodes. The gating module can, to some extent, help the model remember necessary information and ignore worthless ones during the training process, whereas the (Gating w/o) shows a slightly decreased performance.

7.5.6 Imputation Results

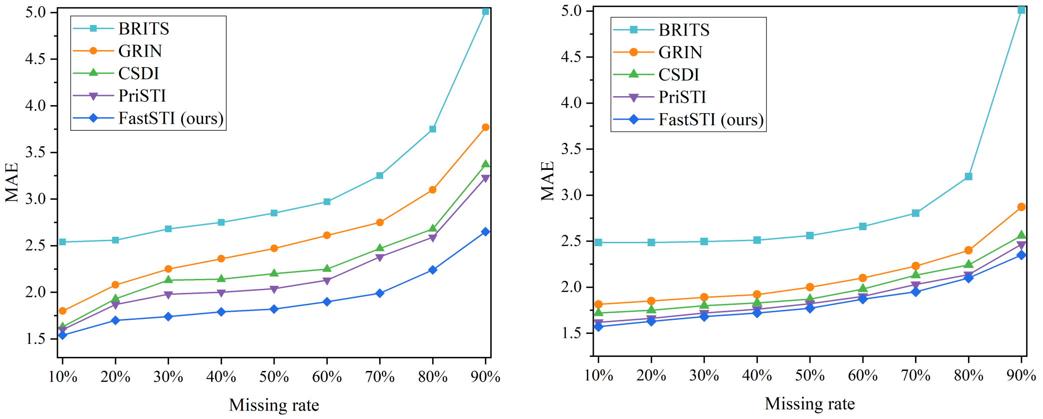
We compare our FastSTI model with sixteen baseline models, including statistical (Mean, KNN [146], Linear InTerPolation (Lin-LTP)), classical ML (MICE [147], Vector AutoRegression (VAR), Kalman Filter (KF)), low-matrix factorization (TRMF [148], BATF [149]), deep autoregressive (BRITS [150], GRIN [136]), and deep generative models (V-RIN [151], GP-VAE [152], rGAIN [153], CSDI [128], SSSD [129], PriSTI [130]) in the missing data imputation domain. Table 7.5 reports the MAE and MSE comparison on METR-LA and PEMS-BAY datasets, while Table 7.6 reports the CRPS metric on the two datasets. FastSTI, using only 6 denoising steps, outperforms all of the baselines, producing more realistic imputation. Focusing on the comparison between FastSTI and PriSTI [130], we have a close but still better performance, proving the effectiveness of the proposed conditional pseudo-numerical method in generating higher-quality samples. FastSTI also benefits from the GCN-based conditional features extractor, in contrast to the message-passing neural network (MPNN) used in PriSTI. Additionally, the comparison between FastSTI-2 and FastSTI-4 favors FastSTI-4, which makes sense, as the imputation quality should increase as the order of the numerical method increases.

Method	METR-LA				PEMS-BAY			
	Block (16.52%)		Point (31.09%)		Block (9.20%)		Point (25.01%)	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Mean	7.48	139.54	7.56	142.22	5.46	87.56	5.42	86.59
KNN	7.79	124.61	7.88	129.29	4.30	49.90	4.30	49.80
Lin-ITP	3.26	33.76	2.43	14.75	1.54	14.14	0.76	1.74
KF	16.75	534.69	16.66	529.96	5.64	93.19	5.68	93.32
MICE	4.22	51.07	4.42	55.07	2.94	28.28	3.09	31.43
VAR	3.11	28.00	2.69	21.10	2.09	16.06	1.30	6.52
TRMF	2.96	22.65	2.86	20.39	1.95	11.21	1.85	10.03
BATF	3.56	35.39	3.58	36.05	2.05	14.48	2.05	14.90
BRITS	2.34	17.00	2.34	16.46	1.70	10.50	1.47	7.94
GRIN	2.03	13.26	1.91	10.41	1.14	6.60	0.67	1.55
V-RIN	6.84	150.08	3.96	49.98	2.49	36.12	1.21	6.08
GP-VAE	6.55	122.33	6.57	127.26	2.86	26.80	3.41	38.95
rGAIN	2.90	21.67	2.83	20.03	2.18	13.96	1.88	10.37
CSDI	1.98	12.62	1.79	8.96	0.86	4.39	0.57	1.12
SSSD	2.95	23.48	2.83	21.95	1.03	7.32	0.97	2.98
PriSTI	1.86	10.70	1.72	8.24	0.78	3.31	0.55	1.03
FastSTI-2	1.81	10.44	1.73	8.17	0.78	3.28	0.51	0.98
FastSTI-4	**1.79**	**10.38**	**1.71**	**8.15**	**0.75**	**3.26**	**0.50**	**0.96**

Table 7.5: MAE and MSE comparison with the baselines (All baseline results are obtained from [130]) [**bold** = best, and underline = second best]

Method	METR-LA		PEMS-BAY	
	Block-M	Point-M	Block-M	Point-M
V-RIN	0.1283	0.7781	0.0394	0.0191
GP-VAE	0.1118	0.0977	0.0436	0.0568
CSDI	0.0260	0.0235	0.0127	0.0067
PriSTI	0.0244	0.0227	**0.0093**	0.0064
FastSTI-2	0.0243	0.0223	0.0095	0.0062
FastSTI-4	**0.0241**	**0.0219**	**0.0093**	**0.0060**

Table 7.6: CPRS comparison (All baseline results are obtained from [130]) [**bold** = best, and underline = second best].



(A) Block-missing scenario

(B) Point-missing scenario

FIGURE 7.9: The impact of missing rate on the imputation performance on METR-LA

We evaluate the impact of missing rates in the METR-LA dataset on the imputation performance in Figures 7.9a and 7.9b. Intuitively, as the rate of missing values increases, data imputation becomes more challenging due to the reduced availability of observed values and the increased complexity of extracting spatiotemporal correlation of traffic conditions. Meanwhile, we can see that our proposed FastSTI-4 consistently outperforms baseline models in terms of imputation performance, regardless of the missing rate. Compared with the best baseline model PriSTI, FastSTI-4 reduces the MAE by up to 17.96% (block-missing at 90% rate) and 7.3% (point-missing at 90% rate).

FastSTI utilizes a tuned variance schedule to improve inference speed in the imputation process, which enables FastSTI to impute highly accurate traffic speed data with only 6 reverse steps (8.3x less than the 50 steps of both PrisTSI [130] and CSDI [128], and 33.3x less than the 200 steps of SSSD [129]). Thereby, FastSTI significantly reduces the computational time required compared to these competing diffusion architectures. Figure 7.10 displays the imputation time of the models for all sensors throughout 2 hours with 5-minute intervals (24 points). Our FastSTI-2 (6 steps) ($\sim 15.12s$) is 5x faster than PriSTI ($\sim 78.96s$) and 4x faster than CSDI ($\sim 60.2s$). The higher-order FastSTI-4 (6 steps) ($\sim 31.44s$) requires more inference time; however, FastSTI-4 (6 steps) is still 2.5x faster than PriSTI and 2x faster than CSDI. Fast-STI (50 steps) requires more time compared to the baseline models due to the utilization of two kinds of high-order pseudo-numerical methods rather than the first-order one, which highly improves imputation accuracy.

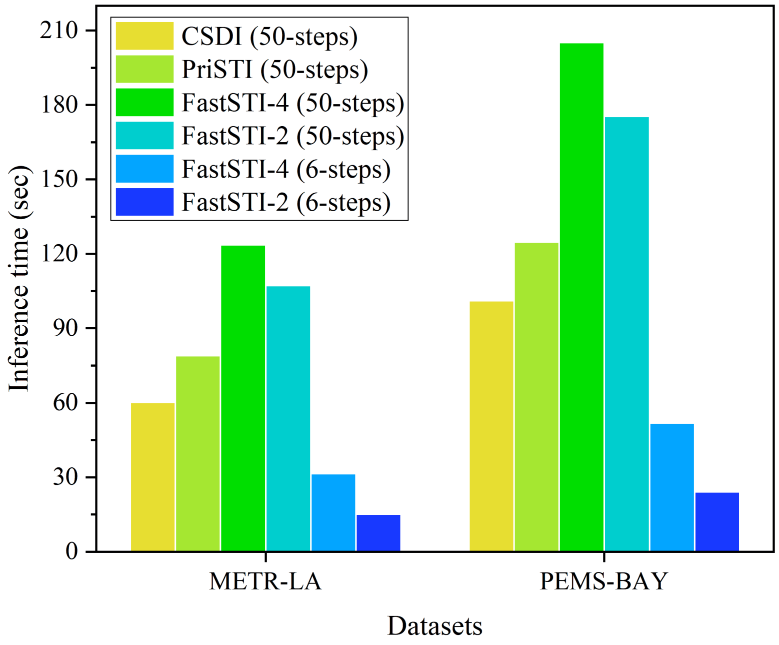


FIGURE 7.10: Inference Times on METR-LA and PEMS-BAY Datasets

PN	Diff-GCN	METR-LA			
		Block-missing		Point-missing	
		MAE	MSE	MAE	MSE
✗	✗	1.86	10.70	1.72	8.24
✗	✓	1.83	10.40	1.74	8.22
✓	✗	1.89	10.87	1.82	8.57
✓	✓	**1.79**	**10.38**	**1.71**	**8.15**

TABLE 7.7: The influence of the different components on FastSTI-4.

7.5.7 Imputation Ablation Experiments

We conduct an ablation study on the METR-LA datasets to verify the impact of different components of our FastSTI model, where Table 7.7 illustrates the performance of FastSTI-4 with and without employing the pseudo-numerical method (PN) and the GCN-based conditional extractor (Diff-GCN). When we remove the pseudo-numerical (PN) methods, the model no longer considers high-order numerical methods converging to the exact solution when Δt is closer to 0. Consequently, it no longer utilizes a larger iteration interval Δt to reduce global error, decreasing imputation accuracy. The spatial learning sub-component, Diff-GCN, is crucial in extracting geographic interactions among nodes within the spatial correlation. Its absence leaves the model with a weak ability to capture the influence among nodes by spreading the traffic feature information on graph G.

Method	METR-LA						PEMS-BAY					
	Block (16.52%)			Point (31.09%)			Block (9.20%)			Point (25.01%)		
	e_1	e_2	e_3	e_1	e_2	e_3	e_1	e_2	e_3	e_1	e_2	e_3
FastSTI-2-50	1.80	10.34	0.0241	1.69	8.01	0.0220	0.72	3.26	91	0.52	1.01	61
FastSTI-4-50	**1.79**	**10.29**	**0.0239**	**1.68**	**7.99**	**0.0219**	**0.70**	**3.24**	**89**	0.51	0.99	**59**
FastSTI-2-6	1.81	10.44	0.0243	1.73	8.17	0.0223	0.78	3.28	95	0.51	0.98	62
FastSTI-4-6	**1.79**	10.38	0.0241	1.71	8.15	**0.0219**	0.75	3.26	93	**0.50**	**0.96**	60

Table 7.8: The impact of increasing the number of steps on FastSTI. e_1 is the MAE, e_2 is the MSE, and e_3 is the CRPS.

Diffsion parameters			Performance	
β_1	β_T	schedule	MAE	MSE
01	0.2	linear	1.96	12.46
		cosine	1.88	11.78
		quadratic	**1.79**	**10.38**
1	0.2	quadratic	2.06	13.87
01	0.1	quadratic	1.86	11.54
	0.3	quadratic	1.91	11.89

Table 7.9: Influence of different diffusion parameters on METR-LA dataset [**bold** = best].

Furthermore, Table 7.8 compares our FastSTI (6 steps) and FastSTI when increasing the number of steps to 50. Surly, increasing the number of steps increases the performance by a noticeable gap, proving the effectiveness of FastSTI quality-wise. However, increasing the number of steps results in a dramatic increase in imputation time.

We further explore the impact of the diffusion parameters on performance using the METR-LA dataset, including the minimum noise level β_1, the maximum noise level β_T, and the diffusion schedule to generate β. Table 7.9 reveals that FastSTI performs best when setting β_1 to 01, β_T to 0.2, and utilizing a quadratic diffusion schedule. Here, the noise level parameter is employed to regulate the diffusion speed. The convergence speed of the model slows down when the β_T is smaller, while the numerical stability of the model is compromised when the β_T is larger. Additionally, regarding the diffusion schedule, a quadratic schedule outperforms linear and cosine schedules, as the quadratic schedule allows for a gentle decay of α_t, improving sample quality and making it the optimal choice for FastSTI.

7.6 Conclusion

We investigated the challenges of traffic speed prediction, proposing the new method DSP-ST. In DSP-ST, we adapt physical and contextual graphs to extract local spatial proximity and global contextual similarities of roads, in addition to an attention Gated-TCN component to learn the temporal dependencies and a GAT component to learn the spatial dependencies of roads. Our experiments on real-world traffic datasets demonstrate our models' ability to improve performance compared to state-of-the-art works in traffic prediction.

Additionally, we investigated the accuracy and speed challenges of traffic data imputation, proposing the FastSTI diffusion model. FastSTI utilizes a higher-order pseudo-numerical methodology for a conditional diffusion model to enhance traffic data imputation accuracy, in addition to utilizing a GCN-based module, serving as feature prior knowledge and capturing the spatiotemporal correlations. Additionally, in FastSTI, we address the time consumption problem of the diffusion models with the utilization of a variance schedule to reduce the sampling iterations. Again, our experiments on real-world traffic datasets proves the effectiveness of our accelerated FastSTI in merging both fast and accurate sampling of the missing traffic data.

7.7 Acknowledgment

The work discussed in this chapter is the result of a collaboration with Shoakang Cheng, a doctoral candidate at the School of Automation, Northwestern Polytechnical University, China, who has served as the primary contributor to this work.

Chapter 8

Anomaly Actions Localization in Semi-Supervised Videos

Chapter Abstract

This chapter details our novel approach and its preliminary results in the extra task of anomaly actions detection and localization in videos, with semi-supervised labeling. We introduce a transformer-based temporal-hierarchical model that weighs the impact of the observed actions in classifying a video as anomalous. Implementing a divide-and-conquer approach over the temporal axis, the video is hierarchically segmented into multiple instances, creating distinct temporal patches. Obtaining sub-predictions from these diverse patches enhances the model's ability to estimate abnormality scores within video segments. [*]

8.1 Introduction

Surveillance plays a central role in almost all security systems; however, extracting important events, particularly anomalies, from the vast pool of collected videos is a time-consuming and exhaustive task. There arises a critical need for an intelligent system capable of accurately and autonomously extracting and localizing events of interest. The main challenge in training such a localization model lies in the lack of supervision, as the massive amount of collected data for this task is

[*] The work in this chapter was carried out during a 3-month visiting period in Computer and Systems Engineering Department, Alexandria University, Egypt, under the supervision of **Prof. Marwan Torki**.

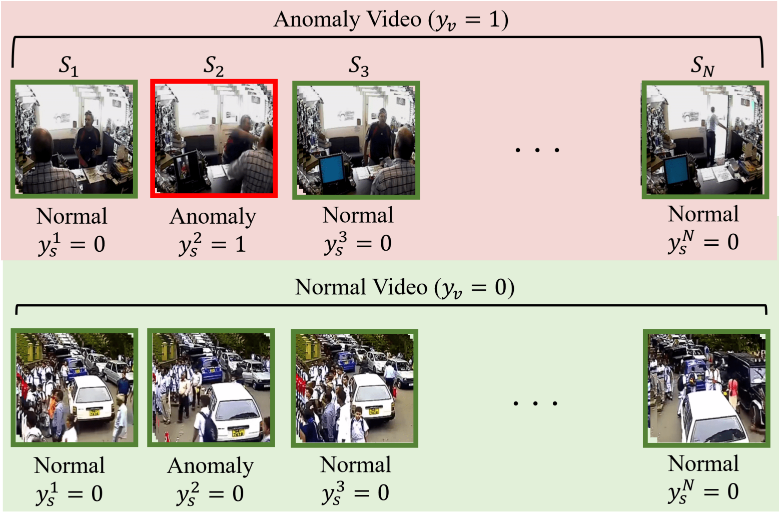


FIGURE 8.1: Each video is split into N segments. A normal video ($y_v = 0$) contains only normal segments ($y_s^i = 0, \forall i \in [1 : N]$). While an anomaly video ($y_v = 1$) contains at least one anomaly segment ($y_s^i = 1, \exists i \in [1 : N]$).

unsupervised or weakly supervised. Consequently, the employed model must be able to decode event sequences, extract embedded relations, and identify potential outlier behaviors. Many prior works have tackled this task, some in a fully unsupervised approach [154,155], but mostly in a semi-supervised approach [24–27,49]. In weakly supervised settings, each video is labeled as normal or anomalous, yet the specific location of the anomaly segment within the anomalous video is unspecified. The standard protocol in the semi-supervised setting operates at the segment level, aiming to maximize the hidden representation gap between normal and anomalous segments, often overlooking per-video classification. However, decoding the relative dependencies within videos should enhance event understanding and improve anomaly localization. Recent attention has been directed towards image reconstruction approaches, wherein models are trained to reconstruct normal videos or frames [156, 157]. Subsequently, these trained models are repurposed to distinguish between high-quality reconstructions, indicative of normal frames, and poor-quality reconstructions, indicative of anomaly frames. In our attempt to address this challenge, we propose a novel approach utilizing per-video classification. If a model can distinguish videos containing anomalies from normal videos, the model's hidden representation should inherently contain sufficient information about the anomaly locations. We proposed a temporal divide-and-conquer transformer-based model to classify the normality of a given video and its segments at various temporal levels. We extract potential anomaly segments based on the aggregated classifications of the model, along with their corresponding activation maps. Preliminary results indicate promising insights into the effectiveness of our proposed approach.

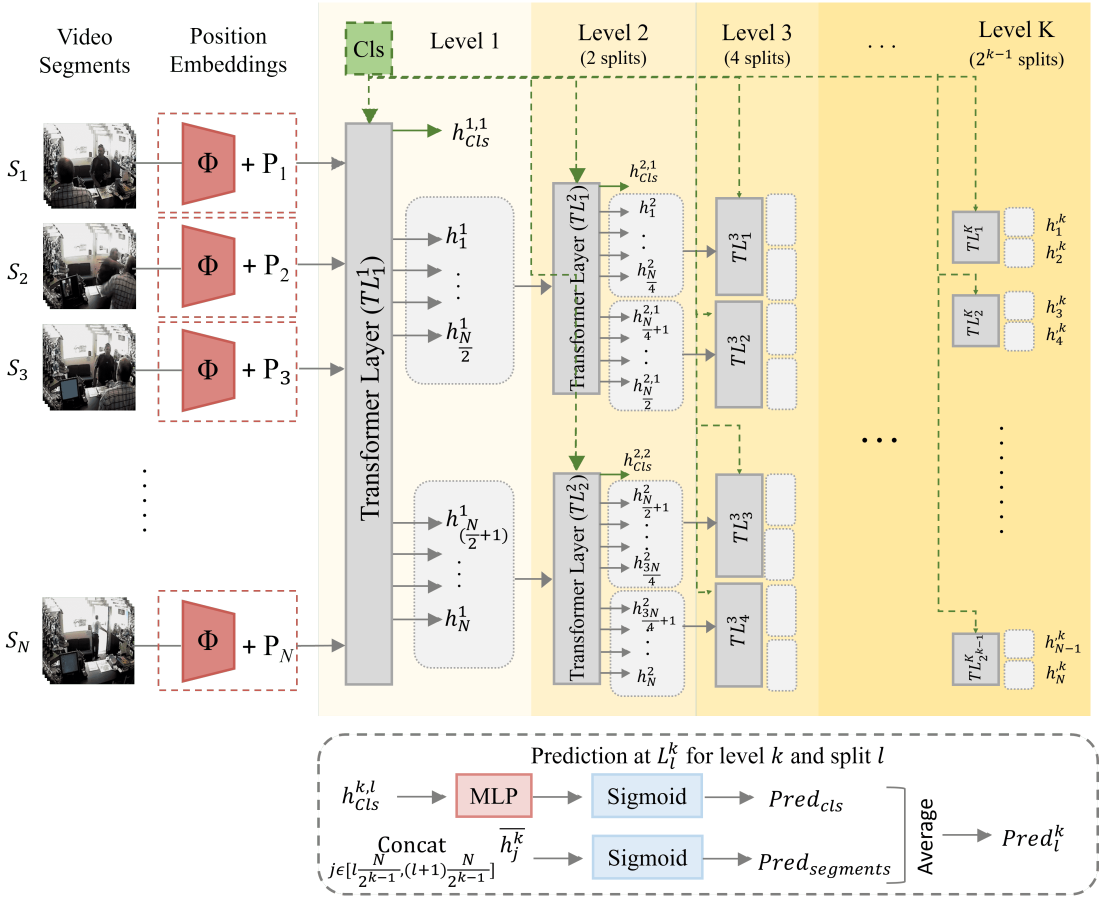


FIGURE 8.2: Our divide-and-conquer transformer-based model

As shown in Figure 8.1, each video is split into a set of segments, where a video is normal if it does not contain any anomalies, while it is an anomaly video if it contains at least one anomaly segment. Unlike previous works, our objective is to solve two tasks: the per-video classification task, predicting y_v, and the per-segment classification, predicting y_s^i for each segment i in a given video. The per-segment classification is fully unsupervised, based on the information extracted during the per-video classification.

8.2 Temporal Divide-and-Conquer Model

We consider two classification tasks: whole video classification and individual segments classification. The input to the model is a whole video, split into a sequence of N segments, where each segment is a set of frames representing an action/event. As depicted in Figure 8.2, the raw images in a segment S_i are projected to the feature space using $\Phi(S_i)$, given in (8.1). Following [27], we extract the features $\mathbf{x}_i \in \mathbb{R}^{1\times D}$ from S_i with a pre-trained SlowFast network [97], where D is the feature size.

$$\mathbf{x}_i = \Phi(S_i) = MLP(SlowFast(S_i)) \tag{8.1}$$

Our divide-and-conquer approach begins with the coarse-grained task of per-video classification, repeatedly breaking down the task into smaller sub-tasks until reaching the fine-grained task of per-segment classification. The input features of the complete sequence $\mathbf{x} \in \mathbb{R}^{N\times D}$ is fed into the first prediction level, as an initial representation of the segments $h_{seg}^{0,1} = (\mathbf{x} + P)$, where P is the position embedding of the segments to preserve the temporal causality. At each prediction level $k \in [1, K]$, where K is the total number of levels, the model encodes the segments signals received from the preceding level $k-1$ with 2^{k-1} parallel self-attention transformer layers $[TL_1^k \dots TL_{2^{k-1}}^k]$. The input to the transformer layer TL_l^k, at level k and split l, is the corresponding segments encoding from the previous level $h_{seg}^{k-1,\lceil \frac{l}{2} \rceil}$ concatenated with the class query Cls, as shown in (8.2).

$$h^{k,l}, w^{k,l} = TL_l^k(Cls \| h_{seg}^{k-1,\lceil \frac{l}{2} \rceil}) \quad \forall l \in [1, 2^{k-1}] \tag{8.2}$$

From each transformer layer TL_l^k, we consider the encoded representations of the segments $h_{seg}^{k,l} \in \mathbb{R}^{\frac{N}{2^{k-1}}\times D}$, the encoded class $h_{cls}^{k,l} \in \mathbb{R}^{1\times D}$, and the attention weights of the class query $w_{cls}^{k,l} \in \mathbb{R}^{1\times N}$, as shown in (8.3).

$$h_{cls}^{k,l} = h^{k,l}[0], \quad h_{seg}^{k,l} = h^{k,l}[1 : \frac{N}{2^{k-1}}], \quad w_{cls}^{k,l} = w^{k,l}[0] \tag{8.3}$$

A split l at level k represents a sub-prediction task of the normality of segments included in the split. To classify the given split l, the model uses both class encoding $h_{cls}^{k,l}$ and segments encoding $h_{seg}^{k,l}$, as in (8.4). To capture the single effect of each segment in the prediction, we apply a single layer perceptron without bias on the average pooled encodings of the segments.

$$\begin{gathered} y_{cls}^{k,l} = Sigmoid(MLP(h_{cls}^{k,l})) \\ y_{seg}^{k,l} = Sigmoid(SLP(AveragePooling(h_{seg}^{k,l}))) \\ y^{k,l} = (y_{cls}^{k,l} + y_{seg}^{k,l})/2 \end{gathered} \tag{8.4}$$

8.3 Localization Approach

Our model breaks the video prediction into a set of sub-predictions, where we aim to measure the influence of a segment S_i in all of its corresponding sub-predictions. To capture such influence, we rely on three measuring factors:

- The abnormality prediction in the corresponding splits across the different levels; the probability of S_i to be an anomaly segment is monotonically increasing with the corresponding probability of a parent split. Therefore, this probability of S_i is measured as the averaged predictions of its parent splits across the K levels ($p_i = \underset{k\in[1,K]}{Average}\ y^{k,l}$, $l = \lceil \frac{i}{2^{k-1}} \rceil$).
- The activation effect of S_i in the prediction y_{seg} of parent splits; The higher is the activation of the segment in an anomaly class, a the higher is its probability of being anomaly. The averaged activation across the levels is given by $a_i = \underset{k\in[1,K]}{Average}\ \overline{h^{k,l}_{seg}}[i]$ and $l = \lceil \frac{i}{2^{k-1}} \rceil$.
- The attention weights of S_i in the class encoding w_{cls}. Again, the higher attention given to a segment during anomaly prediction, the more is its influence and the higher is its probability to be an anomaly. Therefore the attention maps of the class encodings in the corresponding splits are averaged across the levels to provide an estimation of effect of S_i in the prediction. ($w_i = \underset{k\in[1,K]}{Average}\ w^{k,l}_{cls}[i]$, $l = \lceil \frac{i}{2^{k-1}} \rceil$).

Based on these three factors, an aggregated estimation of the abnormality score n_i of a segment is computed as the weighted average of the normalized factors, as given in (8.4), where α, β, and γ are weighting parameters.

$$n_i = \alpha y_i + \beta w_i + \gamma h_i \tag{8.5}$$

Considering the fact that anomaly segments are mostly grouped together in the video, we smooth the estimated probabilities of the segments across the video with moving average followed by spikes filtering to remove possible outlier scores.

8.4 Experimental Results

8.4.1 Dataset

We evaluate our model on the UCF-Crime dataset (see Subsection 1.2.4) [49]. The dataset consists of 1900 surveillance videos, over 128 hours, and contains 13 different anomaly behaviors, (*i.e., Abuse, Assault, accident, fighting, robbery, . . .*). The videos are split into 32 segments, and the ground truth per-segment class is provided for the 32 segments in only the test split of the data.

Following the standard evaluation of the task [24], we use AUC-ROC to evaluate the detection performance of anomalies, and we also report the accuracy and the F1-score.

8.4.2 Implementation details

To increase the temporal resolution of the input, we split the videos into $N = 64$ segments, setting the number of levels to $K = 6$. The feature size D is 288, and each transformer layer includes 8 self-attention heads. The model is trained for 100 epochs with an SGD optimizer with a learning rate 0.01 and binary cross entropy loss. The weighting parameters α, β, and γ are set to 0.9, 0.05, and 0.05, respectively.

8.4.3 Preliminary Results

First, We report the Preliminary outcomes of our approach. As shown in Table 8.1, we divide the SOTA works into two categories: basic multiple instance learning models [24,25,49], and iterative MIL with pseudo labeling model [26–28]. Our model achieves slightly better performance with the best performing basic MIL learning [24], yet it overlooks the bags pairing techniques used in the MIL approaches, achieving good performance by only measuring the influence of the segments in the individual videos classification. However, compared to the iterative techniques, our model achieves degraded performance. Such techniques aim to overcome weak supervision by providing a form of pseudo labels to the training dataset, transforming the problem to fully supervised learning, which explains the out-performance of these techniques.

	Approach	Backbone	AUC
MIL [49]	Inter-bags distance ↑	C3D + FC	75.41
TCN-IBL [25]	Inter-bags ↑ & Intra-bags distance ↓	C3D + TCN	78.66
MIL-MA [24]	Attention-based inter-bags distance ↑	PWCNet + TAN	79.00
GC-LNC [28]	Iterative Label Cleaning	C3D + GCN	82.12
MIL-MIST [26]	Pseudo Labels	I3D + Attention	**82.30**
MIL-Aug [27]	Pseudo Labels	SlowFast + FC	81.24
Ours	Estimate segments activation	SlowFast + Transformer	79.47

Table 8.1: Comparison with the state-of-the-art works.

# of Levels (K)		*Cls*	Multi-scale	ACC	AUC	F1
$K = 1$	$K = 6$	Query	Input			
✓	✗	✗	✗	83.97	83.89	83.09
✓	✗	✓	✗	85.02	84.83	83.40
✓	✗	✓	✓	85.02	84.90	83.89
✗	✓	✓	✓	**87.46**	**87.43**	**87.05**

Table 8.2: Per-video performance ablation.

Then, we show initial ablation experiments evaluating the different components in the proposed approach. Table 8.2 reports the per-video performance with different model variants. Notably, including the sub-predictions levels has the strongest influence on the per-video classification, where breaking down the problem into sub-tasks and attempting to solve these tasks propagates to the primary classification task, improving its performance. Additionally, integrating a class query into the transformer layers adds value to the performance, as it is able to give more attention to the highly related segments to the prediction. Finally, as mentioned earlier, we increase the temporal resolution by increasing the segmentation from the standard 32 to 64; however, integrating both resolutions through averaging into the model allows for higher temporal interpretation and better performance, as shown in the table.

Finally, Table 8.3 illustrates the per-segment anomaly localization using different estimation techniques. Again, the inclusion of multiple levels has the most significant effect on the detection performance, where the score of each segment is measured concerning multiple temporal splits. The activation maps of the segments and the attention weights of the class query provide a reasonable estimation of the segments' abnormality, where this estimation is the best when used together. The model reports a degraded accuracy and F1, indicating an increase in the miss-rate of the normal class. Although, the F1 drops by 2.5% against 5% increase in the AUC, which is the most meaningful metric in the anomaly detection task.

# of Levels (K)		Activation Maps	Attention Weights	Predictions	ACC	AUC	F1
$K = 1$	$K = 6$						
✓	✗	✓	✗	✗	**73.73**	72.83	40.11
✓	✗	✗	✓	✗	70.09	72.28	38.17
✓	✗	✓	✓	✗	73.18	73.87	**40.65**
✗	✓	✓	✓	✓	66.54	**79.47**	37.78

TABLE 8.3: Per-Segment performance ablation.

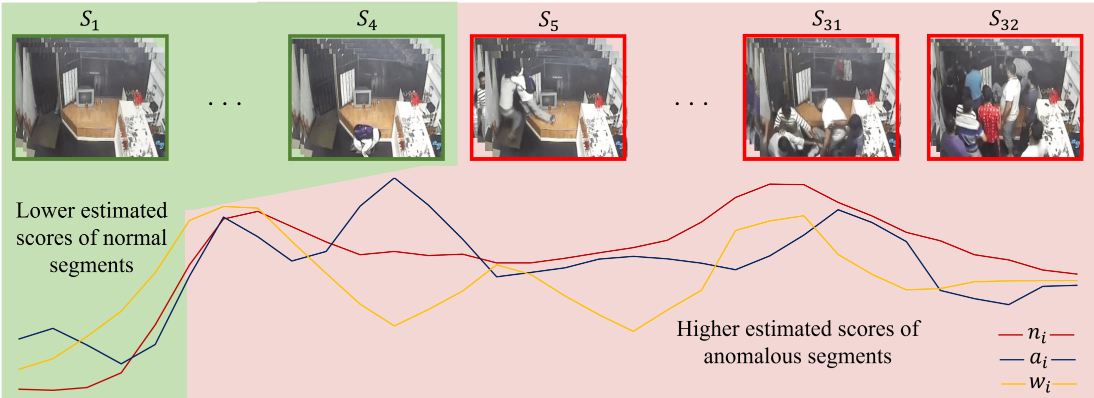


FIGURE 8.3: A qualitative example on the anomaly scores estimations for the anomaly action *"Assult"*, where the action starts at segment S_5 and continues until final segment S_{32}. n_i is the score estimated in (8.5), a_i is the segments activation weights, and w_i is the attention weights.

8.4.4 Qualitative Example

Figure 8.3 plots the estimated anomaly scores of the segments contained in an anomaly video (*"Assult"* event). The video begins with normal events during the first segments $[S_1 - S_4]$; then the anomalous event starts at S_5. As shown, all three abnormality measuring factors report lower scores at the beginning but get higher as the anomaly action starts. The aggregated anomaly estimation n_i gives smooth estimations along the segments, benefiting from the fused estimation factors. The activation map a_i and the attention weights w_i report slightly decreased scores during the anomaly event, yet they are still higher than the normal segments. The anomaly estimation starts to decrease by the end of the video, where the *"assaulting"* event starts to decay. This proves the ability of the model to interpret the events apparent in the segments and distinguish anomaly ones, depending solely on the information extracted during classifying the video.

8.5 conclusion

In this chapter, we detailed our work on the anomaly localization and detection task in semi-supervised videos. The preliminary results provide promising insights into the proposed approach, with a close performance to the state-of-the-art works in the problem. However, the work is still open to further improvements.

Our future direction will aim to tune the predictions of the sub-video-sequences. In our current setting, we assume anomaly labeling for all sub-predictions related to an anomaly video. However, the perfect scenario is to have the correct labels for all predictions. Based on the SOTA comparison, pseudo-labeling allows for better performance, which could be a promising direction. Another promising direction is decoding the single frames, where more information could be extracted from a single anomaly, helping the model better detect the anomaly events.

Chapter 9

Conclusions and Future Work

We conclude the work integrated into this dissertation, summarizing our contributions and findings and exploring the possible directions of future work. In summary, we addressed the human action anticipation task, focusing on two key domains for action anticipation: kitchen activities anticipation and pedestrian action anticipation. Then, we expanded our research to traffic flow anticipation and tackled the anomaly actions detection domain. Our research focused on designing, developing, and evaluating different deep-learning models and approaches across diverse domains.

9.1 Summary of Contributions

SlowFast RULSTM, We introduced a novel multi-time-scale attention-based approach that combines information extracted from varied time scales to anticipate human actions in egocentric videos. Our methodology involved employing two time branches, slow and fast. This dual-branch design facilitated the discrimination of diverse actions with different progressing rates. We investigated several fusion techniques to combine multiple input modalities, demonstrating the model's tendency to benefit from fusing the input modalities prior to integrating the different time scales. Our proposed approach outperformed a state-of-the-art model on two well-known kitchen activities benchmarks, EpicKitchens-55 and EGTEA GAZE+. We also demonstrated superior performance compared to another multi-scale model on the EpicKitchens-55 dataset.

Early Intent Anticipation, We moved to the pedestrian action anticipation task, revisiting the standard anticipation protocol employed for pedestrian intent prediction and expanding the intent anticipation to several seconds before the event. Our proposed early anticipation protocol showed robust performance with no or negligible impact on early anticipation with respect to late anticipation. We adapted RULSTM for intent anticipation, and to further improve the early anticipation performance, we extended the model by incorporating a *goal* module designed to anticipate future features, thereby enhancing the prediction. Our model proved out-performance, particularly in early anticipations, compared to a state-of-the-art model in intent prediction on JAAD and PIE datasets. The integration of the *goal* module results in a notable improvement in prediction accuracy, reaching up to 3% compared to models that do not anticipate future features.

TAMFORMER, Here, we addressed pedestrian action anticipation again, introducing a transformer-based model with a multi-modality approach. We upgraded our transformers with a module that dynamically learns attention masks to measure the correspondences within temporal sequences. We additionally employed a novel loss function, aiming to narrow the performance gap between early anticipation times and the latest anticipation. Our experimental results highlighted the superior performance of our proposed model, showing an improvement of up to +2% F1 on PIE and +5% F1 on JAAD within the $[2-1]$s anticipation range.

Language-Aided Anticipation, Continuing on the pedestrian action anticipation task, we expanded the conventional binary task of predicting the pedestrian's intent to cross or not cross to encompass the anticipation of multiple actions. We then proposed and investigated the incorporation of language descriptions for pedestrian action anticipation. Our investigation involved diverse language generation approaches and tools, proving the efficacy of merging vision-language features to augment the comprehension of visual scenes and elevate anticipation performance. Our evaluation on a novel large-scale urban scenes dataset, named LOKI, proved the superiority of our language-aided anticipation approach, reaching a notable 29.5% F1-score improvement at the 1-second anticipation and 16.66% improvement at the 4-second anticipation, over to not using language.

Traffic Flow Anticipation and Data Imputation, We addressed the problem of traffic behavior anticipation, proposing a dual-graph-based approach that considers the contextual correlations within the road network in addition to the physical neighboring correlation. The proposed model employs an attention-based approach to encode the spatial and temporal information, in addition to a utilized GCN for encoding the dual-graph. We extended the problem to traffic data imputation. The experimental evaluation demonstrated the outperformance of our model compared to the existing prediction baselines.

Anomaly Actions Recognition and Localization, We investigated the task of abnormal behavior recognition in semi-supervised videos, proposing a new transformer-based model for anomalous segments localization, utilizing a multi-level scoring approach. The preliminary results provide promising insights into the proposed approach, with a close performance to the state-of-the-art works in the problem.

9.2 Future work

The significance of human action anticipation is evident in many applications, as explained throughout our work. Despite the current advancements, anticipation performance remains constrained, particularly the early anticipation of multi-second predictions, as deep learning models have yet to match human capabilities in comprehending and perceiving visual stimuli. This limitation prompts the exploration of promising directions for field advancement.

One direction is represented in the integration of language and vision in anticipation tasks. For example, vision question answering could guide image captioners to furnish tailored captions in response to posed questions, opening new dimensions for anticipatory understanding. Another promising direction is the collective anticipation involving multiple agents. This broader scope allows for considering the interactions among agents within an environment during anticipation. For instance, simultaneously predicting the actions of multiple pedestrians while factoring in their mutual influence holds the potential for heightened context comprehension and superior anticipation. Expanding collective anticipation to include vehicles, bicycles, and other potential agents in street scenes promises a more comprehensive understanding. To this extent, our future work will include, but not limited to, the exploration of these directions, investigating their potentials and tailoring models that effectively leverage additional information to augment the anticipation.

Moreover, the anticipation task can be extended to various other applications. For instance, broadening the scope of anomaly detection to include anomaly anticipation could significantly enhance the capabilities of security and surveillance systems, contributing to the prevention of accidents and abnormal behaviors. However, this particular task has not been thoroughly explored in the research community, especially with insufficient grounding for the training data. Therefore, investigating this direction could unveil numerous research opportunities.

Bibliography

[1] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K. L. Koay, and K. Dautenhahn, "Formal verification of an autonomous personal robotic assistant," in *2014 AAAI Spring Symposium Series*, 2014.

[2] M. BARAKAZI, "The use of robotics in the kitchens of the future: The example of'moley robotics'," *Journal of Tourism and Gastronomy Studies*, vol. 10, no. 2, pp. 895–905, 2022.

[3] H. Harman and P. Simoens, "Action graphs for proactive robot assistance in smart environments," *Journal of Ambient Intelligence and Smart Environments*, vol. 12, no. 2, pp. 79–99, 2020.

[4] M. Khoramshahi and A. Billard, "A dynamical system approach for detection and reaction to human guidance in physical human–robot interaction," *Autonomous Robots*, vol. 44, no. 8, pp. 1411–1429, 2020.

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[6] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5725–5734.

[7] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proc. of IEEE/CVF International Conference on Computer Vision Workshops*, 2017.

[8] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[9] "Moley kitchen," https://www.moley.com/.

[10] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, pp. 2259–2322, 2021.

[11] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1366–1401, 2022.

[12] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[13] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Predicting the future: A jointly learnt model for action anticipation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5562–5571.

[14] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3. IEEE, 2004, pp. 32–36.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," 06 2014, pp. 1725–1732.

[16] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," 06 2015, pp. 4694–4702.

[17] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," 06 2015, pp. 2625–2634.

[18] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," 07 2017, pp. 4724–4733.

[19] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," 06 2018, pp. 6450–6459.

[20] W. Graham, R. Fergus, Y. Lecun, and C. Bregler, "Convolutional learning of spatio-temporal features," vol. 6316, 12 2010.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 12 2015, pp. 4489–4497.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 1, 06 2014.

[23] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," 04 2016.

[24] Y. Zhu and S. Newsam, "Motion-aware feature for improved video anomaly detection," *arXiv preprint arXiv:1907.10211*, 2019.

[25] J. Zhang, L. Qing, and J. Miao, "Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection," in *2019 IEEE International Conference on Image Processing (ICIP).* IEEE, 2019, pp. 4030–4034.

[26] J.-C. Feng, F.-T. Hong, and W.-S. Zheng, "Mist: Multiple instance self-training framework for video anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 009–14 018.

[27] K. El-Tahan and M. Torki, "Semi-supervised anomaly detection for weakly-annotated videos." in *VISIGRAPP (5: VISAPP)*, 2022, pp. 871–878.

[28] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1237–1246.

[29] M. Raptis and L. Sigal, "Poselet key-framing: A model for human activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2650–2657.

[30] M. Hoai and F. De la Torre, "Max-margin early event detectors," *International Journal of Computer Vision*, vol. 107, pp. 191–202, 2014.

[31] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *2011 international conference on computer vision.* IEEE, 2011, pp. 1036–1043.

[32] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13.* Springer, 2014, pp. 596–611.

[33] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, and S. Wang, "Recognize human activities from partially observed videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2658–2665.

[34] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12.* Springer, 2012, pp. 286–299.

[35] F. Becattini, T. Uricchio, L. Seidenari, L. Ballan, and A. D. Bimbo, "Am i done? predicting action progress in videos," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 4, pp. 1–24, 2020.

[36] M. Sadegh Aliakbarian, F. Sadat Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging lstms to anticipate actions very early," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 280–289.

[37] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1942–1950.

[38] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1473–1481.

[39] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13.* Springer, 2014, pp. 689–704.

[40] Y. Kong, S. Gao, B. Sun, and Y. Fu, "Action prediction from videos via memorizing hard-to-predict samples," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[41] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," 07 2017.

[42] A. Furnari and G. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[43] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[44] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 639–655.

[45] "Epic-kitchens," https://epic-kitchens.github.io/2023.

[46] H. Girase, H. Gang, S. Malla, J. Li, A. Kanehara, K. Mangalam, and C. Choi, "Loki: Long term and key intentions for trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9803–9812.

[47] "Pie dataset," https://data.nvision2.eecs.yorku.ca/PIE_dataset/.

[48] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[49] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[50] N. Osman, G. Camporese, P. Coscia, and L. Ballan, "SlowFast Rolling-Unrolling LSTMs for action anticipation in egocentric videos," in *Proc. of the IEEE/CVF International Conference on Computer Vision Workshops*, 2021.

[51] N. Osman, E. Cancelli, G. Camporese, P. Coscia, and L. Ballan, "Early pedestrian intent prediction via features estimation," in *IEEE ICIP*, 2022, pp. 3446–3450.

[52] N. Osman, G. Camporese, and L. Ballan, "Tamformer: Multi-modal transformer with learned attention mask for early intent prediction," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[53] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what? - anticipating temporal occurrences of activities," 06 2018.

[54] P. Felsen, P. Agrawal, and J. Malik, "What will happen next? forecasting player moves in sports videos," 10 2017, pp. 3362–3371.

[55] T. Mahmud, M. Hasan, and A. Roy-Chowdhury, "Joint prediction of activity labels and starting times in untrimmed videos," 10 2017.

[56] K.-H. Zeng, W. Shen, D.-A. Huang, M. Sun, and J. C. Niebles, "Visual forecasting by imitating dynamics in natural sequences," 08 2017.

[57] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3118–3125.

[58] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for evaluating pedestrian action prediction," in *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[59] C. Fan, J. Lee, and M. Ryoo, "Forecasting hand and object locations in future frames," 05 2017.

[60] A. Furnari, S. Battiato, K. Grauman, and G. Farinella, "Next-active-object prediction from egocentric videos," *Journal of Visual Communication and Image Representation*, vol. 49, 10 2017.

[61] N. Rhinehart and K. Kitani, "First-person activity forecasting with online inverse reinforcement learning," 10 2017.

[62] M. Zhang, K. Ma, J. Lim, Q. Zhao, and J. Feng, "Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks," 07 2017, pp. 3539–3548.

[63] G. Camporese, P. Coscia, A. Furnari, G. Farinella, and L. Ballan, "Knowledge distillation for action anticipation via label smoothing," 01 2021, pp. 3312–3319.

[64] S. Neogi, M. Hoy, K. Dang, H. Yu, and J. Dauwels, "Context model for pedestrian intention prediction using factored latent-dynamic conditional random fields." *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6821–6832, 2020.

[65] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Do they want to cross? understanding pedestrian intention for behavior prediction." in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[66] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[67] D. Yang, H. Zhang, E. Yurtsever, K. Redmill, and Ü. Özgüner, "Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention." *arXiv preprint arXiv:2104.05485*, 2021.

[68] J. Lorenzo, I. P. Alonso, R. Izquierdo, A. L. Ballardini, Á. H. Saz, D. F. Llorca, and M. Á. Sotelo, "CAPformer: pedestrian crossing action prediction using transformer." *Sensors*, vol. 21, no. 17, 2021.

[69] J. Gesnouin, S. Pechberti, B. Stanciulcscu, and F. Moutarde, "Rnn-based pedestrian crossing prediction using activity and pose-related features," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, 2021.

[70] ——, "TrouSPI-Net: Spatio-temporal attention on parallel atrous convolutions and u-grus for skeletal pedestrian crossing prediction," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, 2021.

[71] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in *2013 American Control Conference.* IEEE, 2013, pp. 4724–4729.

[72] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[73] P. Theja and L. Vanajakshi, "Short term prediction of traffic parameters using support vector machines technique," in *2010 3rd International Conference on Emerging Trends in Engineering and Technology.* IEEE, 2010, pp. 70–75.

[74] J. Ahn, E. Ko, and E. Y. Kim, "Highway traffic flow prediction using support vector regression and bayesian classifier," in *2016 International Conference on Big Data and Smart Computing (BigComp).* IEEE, 2016, pp. 239–244.

[75] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” *Expert Systems with Applications*, p. 117921, 2022.

[76] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors*, vol. 17, no. 7, p. 1501, 2017.

[77] X. Ma, H. Zhong, Y. Li, J. Ma, Z. Cui, and Y. Wang, “Forecasting transportation network speed using deep capsule networks with nested lstm models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4813–4824, 2020.

[78] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[79] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, “Temporal multi-graph convolutional network for traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3337–3348, 2020.

[80] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” *arXiv preprint arXiv:1906.00121*, 2019.

[81] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the dots: Multivariate time series forecasting with graph neural networks,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[82] J. Qi, Z. Zhao, E. Tanin, T. Cui, N. Nassir, and M. Sarvi, “A graph and attentive multi-path convolutional network for traffic prediction,” *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[83] H. Wu and D. Levinson, “The ensemble approach to forecasting: a review and synthesis,” *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103357, 2021.

[84] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, “Lc-rnn: A deep learning model for traffic speed prediction.” in *IJCAI*, vol. 2018, 2018, p. 27.

[85] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. M. Rahman, “Unified spatio-temporal modeling for traffic forecasting using graph neural network,” in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[86] Y. Shin and Y. Yoon, "Pgcn: Progressive graph convolutional networks for spatial-temporal traffic forecasting," *arXiv preprint arXiv:2202.08982*, 2022.

[87] Y. Wang, "Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1s, pp. 1–25, 2021.

[88] X. Hu, J. Dai, M. Li, C. Peng, Y. Li, and S. Du, "Online human action detection and anticipation in videos: A survey," *Neurocomputing*, vol. 491, pp. 395–413, 2022.

[89] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multitask recurrent neural network for online action detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2667–2682, 2018.

[90] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1439–1449.

[91] S. Nah, T. Kim, and K. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," 07 2017, pp. 257–265.

[92] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang, and S.-F. Chang, "Multi-modal multiscale deep learning for large-scale image annotation," *IEEE Transactions on Image Processing*, vol. 28, pp. 1720–1731, 04 2019.

[93] N. Van Noord and E. Postma, "Learning scale-variant and scale-invariant features for deep image classification," *Pattern Recognition*, vol. 61, pp. 583–592, 2017.

[94] S. Hao, Y. Cui, and J. Wang, "Segmentation scale effect analysis in the object-oriented method of high-spatial-resolution image classification," *Sensors*, vol. 21, no. 23, p. 7935, 2021.

[95] V. S. Martins, A. L. Kaleita, B. K. Gelder, H. L. da Silveira, and C. A. Abe, "Exploring multiscale object-based convolutional neural network (multiocnn) for remote sensing image classification at high spatial resolution," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 168, pp. 56–73, 2020.

[96] J. Wang, Y. Zheng, M. Wang, Q. Shen, and J. Huang, "Object-scale adaptive convolutional neural networks for high-spatial resolution remote sensing

image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 283–299, 2020.

[97] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[98] F. Sener, D. Singhania, and A. Yao, *Temporal Aggregate Representations for Long-Range Video Understanding*, 10 2020, pp. 154–171.

[99] A. Burns, R. Tan, K. Saenko, S. Sclaroff, and B. A. Plummer, "Language features matter: Effective language representations for vision-language tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7474–7483.

[100] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[101] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning.* PMLR, 2021, pp. 8748–8763.

[102] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[103] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning.* PMLR, 2022, pp. 12 888–12 900.

[104] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[105] V. Manousaki, K. Bacharidis, K. Papoutsakis, and A. Argyros, "Vlmah: Visual-linguistic modeling of action history for effective action anticipation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1917–1927.

[106] Q. Zhao, C. Zhang, S. Wang, C. Fu, N. Agarwal, K. Lee, and C. Sun, "Antgpt: Can large language models help long-term action anticipation from videos?" *arXiv preprint arXiv:2307.16368*, 2023.

[107] S. Ghosh, T. Aggarwal, M. Hoai, and N. Balasubramanian, “Distilling knowledge from language models for video-based action anticipation,” *arXiv preprint arXiv:2210.05991*, 2022.

[108] D. Huang, O. Hilliges, L. Van Gool, and X. Wang, “Palm: Predicting actions through language models@ ego4d long-term action anticipation challenge 2023,” *arXiv preprint arXiv:2306.16545*, 2023.

[109] S. Das and M. S. Ryoo, “Video+ clip baseline for ego4d long-term action anticipation,” *arXiv preprint arXiv:2207.00579*, 2022.

[110] D. Damen, H. Doughty, G. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, “The epic-kitchens dataset: Collection, challenges and baselines,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[111] A. Furnari, S. Battiato, and G. M. Farinella, “Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation,” in *International Workshop on Egocentric Perception, Interaction and Computing (EPIC) in conjunction with ECCV*, 2018.

[112] C. Vondrick, H. Pirsiavash, and A. Torralba, “Anticipating visual representations from unlabeled video,” 06 2016, pp. 98–106.

[113] J. Munro and D. Damen, “Multi-modal Domain Adaptation for Fine-grained Action Recognition,” in *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[114] S. Song, J. Liu, Y. Li, and Z. Guo, “Modality compensation network: Cross-modal adaptation for action recognition,” *IEEE Transactions on Image Processing*, vol. 29, pp. 3957–3969, 2020.

[115] X. Wang, Y. Wu, L. Zhu, and Y. Yang, “Symbiotic attention with privileged information for egocentric action recognition,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 249–12 256, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6907

[116] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information*

Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[117] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *ICCV*, 2019.

[118] I. Kotseruba and A. Rasouli, "Benchmark for evaluating pedestrian action prediction," in *WACV*, 2021.

[119] A. Furnari and G. M. Farinella, "What Would You Expect? Anticipating Egocentric Actions With Rolling-Unrolling LSTMs and Modality Attention," in *ICCV*, 2019.

[120] G. Guo, W. Yuan, J. Liu, Y. Lv, and W. Liu, "Traffic forecasting via dilated temporal convolution with peak-sensitive loss," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 1, 2023.

[121] W. Zhang, F. Zhu, Y. Lv, C. Tan, W. Liu, X. Zhang, and F.-Y. Wang, "Adapgl: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 139, p. 103659, 2022.

[122] H. Gülaçar, Y. Yaslan, and S. F. Oktuğ, "Short term traffic speed prediction using different feature sets and sensor clusters," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium.* IEEE, 2016, pp. 1265–1268.

[123] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3700–3709, 2018.

[124] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[125] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *International Conference on Learning Representations*, 2021.

[126] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning.* PMLR, 2021, pp. 8162–8171.

[127] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *arXiv preprint arXiv:2209.00796*, 2022.

[128] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.

[129] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *Transactions on Machine Learning Research*, 2022.

[130] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "Pristi: A conditional diffusion framework for spatiotemporal imputation," *International Conference on Data Engineering*, 2023.

[131] Y. Duan, N. Chen, S. Shen, P. Zhang, Y. Qu, and S. Yu, "Fdsa-stg: Fully dynamic self-attention spatio-temporal graph networks for intelligent traffic flow prediction," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9250–9260, 2022.

[132] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data," in *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, 2020, pp. 3074–3082.

[133] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: Long short-term traffic prediction with graph convolutional networks." in *IJCAI*, vol. 7, 2020, pp. 2355–2361.

[134] E. J. Keogh and M. J. Pazzani, *Derivative Dynamic Time Warping*, pp. 1–11. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611972719.1

[135] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, USA:, 1994, pp. 359–370.

[136] A. Cini, I. Marisca, and C. Alippi, “Filling the g_ap_s: Multivariate time series imputation by graph neural networks,” *International Conference on Learning Representations*, 2022.

[137] B. Yang, Y. Kang, Y. Yuan, X. Huang, and H. Li, “St-lbagan: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation,” *Knowledge-Based Systems*, vol. 215, p. 106705, 2021.

[138] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, “Pseudo numerical methods for diffusion models on manifolds,” *International Conference on Learning Representations*, 2022.

[139] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” *arXiv preprint arXiv:2009.09761*, 2020.

[140] C. Tian and W. K. Chan, “Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies,” *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 549–561, 2021.

[141] J. E. Matheson and R. L. Winkler, “Scoring rules for continuous probability distributions,” *Management science*, vol. 22, no. 10, pp. 1087–1096, 1976.

[142] C.-H. Wu, J.-M. Ho, and D.-T. Lee, “Travel-time prediction with support vector regression,” *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[143] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” *arXiv preprint arXiv:1709.04875*, 2017.

[144] C. Zheng, X. Fan, C. Wang, and J. Qi, “Gman: A graph multi-attention network for traffic prediction,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[145] C. Shang, J. Chen, and J. Bi, “Discrete graph structure learning for forecasting multiple time series,” *arXiv preprint arXiv:2101.06861*, 2021.

[146] L. Beretta and A. Santaniello, “Nearest neighbor imputation algorithms: a critical evaluation,” *BMC medical informatics and decision making*, vol. 16, no. 3, pp. 197–208, 2016.

[147] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: issues and guidance for practice," *Statistics in medicine*, vol. 30, no. 4, pp. 377–399, 2011.

[148] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," *Advances in neural information processing systems*, vol. 29, 2016.

[149] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[150] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[151] A. W. Mulyadi, E. Jun, and H.-I. Suk, "Uncertainty-aware variational-recurrent imputation network for clinical time series," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9684–9694, 2021.

[152] V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt, "Gp-vae: Deep probabilistic time series imputation," in *International conference on artificial intelligence and statistics.* PMLR, 2020, pp. 1651–1661.

[153] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning.* PMLR, 2018, pp. 5689–5698.

[154] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly detection in video via self-supervised and multi-task learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 742–12 752.

[155] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, "Appearance-motion memory consistency network for video anomaly detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 938–946.

[156] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 183–14 193.

[157] S. Bhakat and G. Ramakrishnan, “Anomaly detection in surveillance videos,” in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2019, pp. 252–255.