UNIVERSITY OF EXETER

Computer Science Department

B.Sc. Computer Science
Computer Science Department

Exploring egocentric action anticipation using RU-LSTMs.

Candidate

Oliver Cieplinski

Student ID 700045551

Supervisor

Dr. Sareh Rowlands

University of Exeter

Co-supervisor

Academic Year
2022/2023

Abstract

The aim of this project is to explore the field of egocentric action anticipation through the baseline architecture of Rolling Unrolling Long Short-Term Memory (RU-LSTM). Firstly, various aspects of RU-LSTM approach, such as sequence completion pre-training and encoding time, are investigated to get an understanding of their contribution to overall performance. Secondly, the accuracy at longer time scales is explored. Subsequently, Anticipative Video Transformer (AVT) is studied as an example of state of the art approaches and its performance dependence on some features, such as encoding time and number of attention heads, explored. It is further investigated whether novel aspects of AVT, such as use of transformer decoder for temporal modelling and its loss function, can be incorporated into RU-LSTM to improve its performance. All the models are trained and validated on the 2022 EPIC-KITCHENS 55 dataset and, due to its high computational requirements, a subset thereof; some experiments are also run on EGTEA-Gaze+ datasaet. It is found that the principal advantage of AVT architecture comes from replacing the core of the RU-LSTM model and thus it was not beneficial to incorporate its features into RU-LSTM. Some promising areas for further study are also identified.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

R-LSTM Rolling Long Short-Term Memory

U-LSTM Unrolling Long Short-Term Memory

RU-LSTM Rolling Unrolling Long Short-Term Memory

AVT Anticipative Video Transformer

GCP Google Cloud Platform

TtA Time to Action

SCP Sequence Completion Pre-Training

GPT Generative Pre-Trained Transformer

EK-55 EPIC-KITCHENS-55

1in10 Dataset consisting of 10% of EK-55 dataset

TSN Temporal Segment Network

SGD Stochastic Gradient Descent

LMDB Lightning Memory-Mapped Database

1 Project Outline

1.1 Project Specification

Egocentric action anticipation is a burgeoning concept within the field of artificial intelligence that tries to predict the future actions of the agent utilising neural networks. Egocentric vision is a field of computer vision, which is specialised on analysing image and video data captured by a camera wearer. This camera will aim to try and approximate what the visual perspective of the camera wearer is, to as high a degree of accuracy as possible. The objects and actions within view of the agent are processed using Human Action Recognition within a first person lens in order to identify and analyse the landscape. These objects are then understood by the processor, and it will try to predict what actions the agent will perform and the future state of the scene within an allotted time period using neural networks and other associated technologies. Action anticipation is an extremely malleable and useful technology, with applications that can stretch further than the field it is housed in. Action anticipation can deal with predicting consecutive activities, identifying and understanding the intentions of humans, and also autonomous driving. The main benefit that action anticipation grants is predicting a future action, whilst only having a portion of the prior information. Combining the aspects of egocentric vision with action anticipation, we have the concept of egocentric action anticipation. The main goal of egocentric action anticipation is to be able to predict the future state of a situation based only off of a portion of the current situation, through the lens of a camera that mimics human vision. We first take the observed action, and then using semantic labels we are able to create probabilities for the most likely future outcome that may proceed. Adding the egocentric element to action anticipation allows for more tangible real-life scenarios, where the actions being analysed are subject to factors such as human error, environmental contributors and spatial awareness. These scenarios within the project present themselves as first person cooking videos, where an agent is carrying out a series of ordered actions on objects within the frame of the scene.

1.2 Motivation

The motivation for this project is the importance of egocentric action anticipation. It has numerous practical applications including enhancing user experience in augmented reality applications where anticipating user's actions allows the system to provide better information and feedback. Another example is the development of more intuitive and responsive robots by enabling them to anticipate people's actions. Some of the other applications are assisting people with limited mobility, analysis of sports video for training purposes, and visual surveillance.

The second motivation is the recent advances in solving the problem of egocentric action anticipation thanks to the recent rapid advances in neural computations, in particular architectures such as RU-LSTM and AVT.

1.3 Aims

We aim to ascertain how the different aspects of the dual neural network setup can be examined and improved upon, and how the impact of these aspects affects the performance of the model. As a result of this experimentation, we hope to be able to positively influence the future research in the field of action anticipation. Alongside dissecting the integral features of the architecture, we also aim to explore how newer architectures within the area can augment the capabilities of the RU-LSTM architecture. Through analysis and experimentation with this new technology e.g. AVT, we hope to be able to bolster RU-LSTM with these newer developments, as well as identifying areas that RU-LSTM is stronger and weaker within. Additionally, we aim to enquire into how the attention mechanisms presented within AVT can offer enhancement to the RU-LSTM architecture.

Specifically, to develop and enhance the RU-LSTM architecture, we aim for a few pertinent goals. Firstly we would like to explore how the model works and how different aspects of it contribute to overall performance. Secondly we aim to try to enhance some of its components by modifying the general attributes of the neural networks. Further we will attempt to seek improvements by exploration changes in the RU-LSTM architecture. Finally we will focus our attention on AVT architecture with the objective of understanding it, exploring how its features contribute to the overall success and whether benefits can be obtained by combining the salient features of AVT or other latest techniques with RU-LSTM baseline.

1.3.1 Success Criteria

For my success criteria, I have based my metrics off of analysis of similar literature to create feasible criterion for the project. The criterion are assembled around the project's theme of developing the RU-LSTM architecture.

- Investigate the main features of the RU-LSTM and AVT architecture
- Augment and develop the features to improve accuracy
- Integrate aspects and mechanisms from other state of the art architectures

2

Design and Implementation

2.1 Datasets

Table 2.1 presents the characteristics of the three main datasets this project identified as the most popular within the field of action anticipation.

	EGTEA Gaze+	Ego4D	EPIC-KITCHENS
Length	*28 Hours*	*3670 Hours*	*100 Hours*
Video Type	*Wearable Gaze Tracking*	*Various*	*Head Mounted Camera*
Contract	*Free*	*Paid*	*Free*
Extra Notes	*Frame level annotations, annotated hand masks*	*Varied and different activities*	*Pause-and-talk narration*

Table 2.1: Comparison of action anticipation video datasets.

A prominent dataset within action anticipation is the EGTEA Gaze+ dataset from Georgia Tech [12]. It has egocentric first person video and hones in on the annotation and detail of the actions occurring, and as such contains action annotations and hand masks. However, it has a far lower quantity of video available compared to the other datasets. It uses gaze technology which can identify where the agent's gaze is directed towards, adding extra information that neural networks can use. These features are very useful towards egocentric action anticipation, as gaze changes provide clues about the upcoming action. In this investigation we focused on video features and did not take advantage of gaze information.

The second dataset is the Ego4D dataset, which is a very large dataset of video and is very well annotated. However, the fact that the video is very varied, as there are a far larger amount of contributors, and the service requires payment, make it less attractive.

The final dataset, is the EPIC-KITCHENS dataset [4]. This dataset comes in two versions of 55 and 100 hours of head mounted egocentric video, with the addition of pause and talk

narration. It consists of first person perspective video which details an agent preparing a meal. The actions are carried out one after another, making them invaluable for egocentric action anticipation analysis. These videos are useful not just for testing the voracity of a neural network, but also training the network due to the large scale of the dataset. The creators of the dataset highlight the importance of explicit temporal modelling, and discussing the granularity of certain actions that may appear unclear [5]. This dataset is thus very well optimised for egocentric action anticipation, with several papers being written using it as a benchmark, e.g. [13].

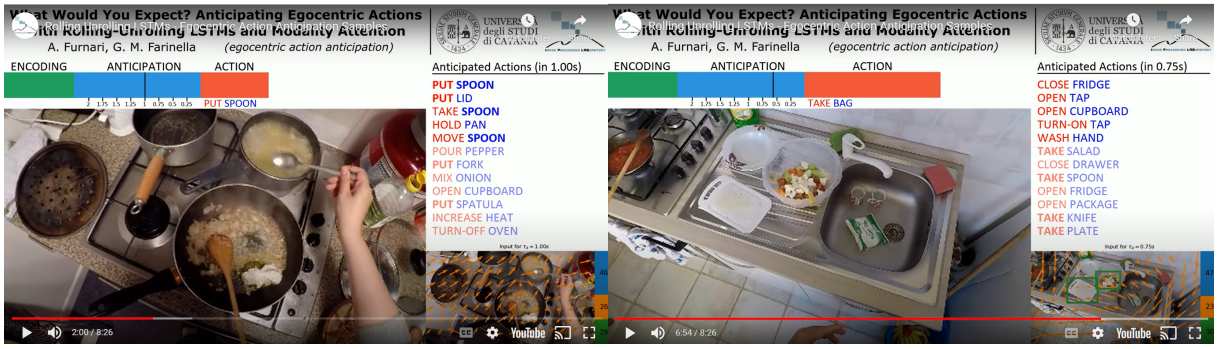

Figure 2.1: Sample EK-55 dataset video frames with annotations from [6].

Due to the limited computational resources and storage available for this work, we decided that the main proportion of the project would be carried out upon the EPIC-KITCHENS datasets. This also promotes homogeneity within the results, and it allows for better comparison of data between the results garnered from the project. Using primarily one dataset also allows for more precise subset preparation and further investigation into certain aspects of the dataset, such as action labels and annotation within the video clips as well. In order to gain more general insight and provide further validation of the tests, some experiments were also carried out on EGTEA-Gaze+ dataset as discussed in detail below.

2.1.1 Details of Use of EPIC-KITCHENS

There are two different sizes of this dataset, EPIC-KITCHENS-55 and EPIC-KITCHENS-100, which offer 55 and 100 hours of video clips respectively. After studying both of these, we decided that the EPIC-KITCHENS-55 (EK-55) dataset is a better choice, as the smaller dataset affords more flexibility in creating subsets of the dataset, as well as yielding faster training and validation speeds and reduced storage requirements.

Even for EK-55, the original videos in the dataset require 1.1TB of storage, which is more than we had available. We were therefore further limited to use the features pre-extracted using Temporal Segment Network (TSN) for RGB, FLOW and OBJ modalities by the authors of [7] and [8] and kindly made available from [6]. The structure of the data within the pre-extracted dataset uses Lightning Memory-Mapped Database (LMDB) format. This format is a high performance key-value store which uses memory-mapping to allow the application to access data directly from memory, meaning data doesn't need to be copied from the dataset to

the software. Each modality possess an LMDB file, which contains a feature vector for each video clip in the dataset. The size of the feature vector is 1024 for RGB and OBJ and 352 for OBJ modality. These files are the raw data in which the program engages its training and validation upon.

The dataset also contains a series of CSV files which identify the video clips for training and validations and provide annotations for objects, verbs and actions. The files encapsulated in the EK-55 dataset are listed in Appendix 5.2.1.

Due to limited computational resources available we found it necessary to prepare a smaller Dataset consisting of 10% of EK-55 dataset (1in10). The details of this subset are described in Appendix 5.2.2.

2.1.2 Details of Use of EGTEA-Gaze+

We performed some experiments on EGTEA Gaze+ dataset `https://cbs.ic.gatech.edu/fpv/`. The original frames were used in this case as just enough storage was available on the GCP virtual machine with GPU to store them and it allowed for limited experimentation with the backbone network as well as head network. We discovered that the RAM available only allowed for processing of up to 6 observation frames, which limited the scope of experiments we were able to perform, particularly on the aspects related to temporal features such as multi-head attention in the head network.

2.2 Architecture Exploration

The next core component that the project will hone in on is exploring different architectures within the field of egocentric action anticipation. Within this sector of the project, the main aim is to attempt to integrate state of the art techniques within the RU-LSTM architecture. Through this exploration, we first identified the AVT architecture as a particular interest. Using the AVT's breakthrough techniques with attention mechanisms, we decided that trying to adopt these techniques would be a useful path to aid with one of the RU-LSTM architecture's biggest issues, which is variable-length dependency. Our design philosophy is based around these tenets, and trying to investigate how these newer features could be combined with best features of RU-LSTM. The success criteria for this part of the project will be the successful evaluation of both architectures, breadth of the coverage of their different aspects, combination of the features and performance improvement measured using the same criteria as for feature experimentation in Section 2.3.

2.2.1 RU-LSTM

The first method we experiment with is RU-LSTM [8], which we described in detail in [3]. The first aspect of the architecture that we investigated was one of the key features cited in the paper by Antonino Furnari [7], which is Sequence Completion Pre-Training (SCP). This

technique pre-trains the two key neural networks within the architecture Rolling Long Short-Term Memory (R-LSTM) and Unrolling Long Short-Term Memory (U-LSTM); during SCP, the parameters of these networks are modified. This allows the U-LSTM to process future representations during training, while the R-LSTM only encodes past representations. The way this is achieved is by sampling input representations from future time-steps for the U-LSTM during pre-training. This means that it can use input data from later time steps. However, the R-LSTM only gets input data from earlier time steps, meaning it can focus on encoding past observations.

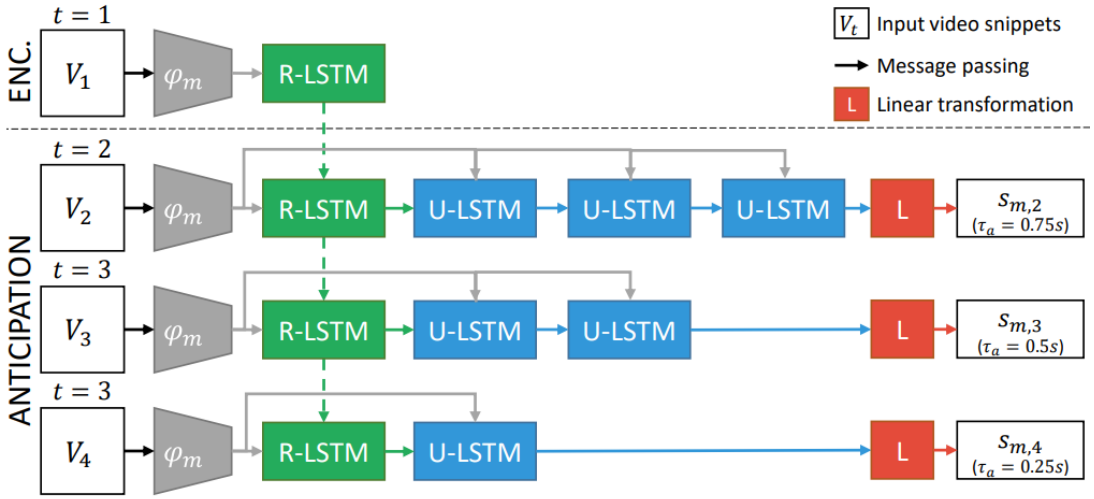

Figure 2.2: RU-LSTM diagram for a single modality from [7].

Next we investigate the anticipation at longer time frames than tested in [7] and generally used in EPIC-KITCHENS challenge. While the expectation is that the performance will deteriorate, we are interested in seeing if the model is robust to going beyond the 3.5s originally tested. In order to conduct this investigation we added an option to RU-LSTM software to specify a time offset in seconds. E.g. a time offset of 1s means that instead of using the last 3.5s of video for observation and anticipation, the period from 4.5s to 1.0s before the start of the anticipated action is used. This approach means that it is possible to use the existing labelling of sequences in actions. It also means that it is possible that for some sequences not enough earlier video is available to train the model, simply because the actions occur too early in the full video. The details on the number of dropped clips are discussed in Appendix 5.2.3; in general we conclude that the numbers are small enough to allow for meaningful testing.

2.2.2 AVT

Transformer neural network architectures introduced in [17] have recently shown very promising results for a variety of tasks including machine translation and language modelling including Generative Pre-Trained Transformer (GPT) [14]. The principal idea of this approach is to use positional encoding and attention to focus the learning in the network on positions in the input sequence which are related. The main advantage of this compared to recurrent

architectures such RU-LSTM is that it allows parallel processing at different positions in the sequence and makes it easier to model long term relationships as attention mechanism can link inputs in any positions in the sequence directly.

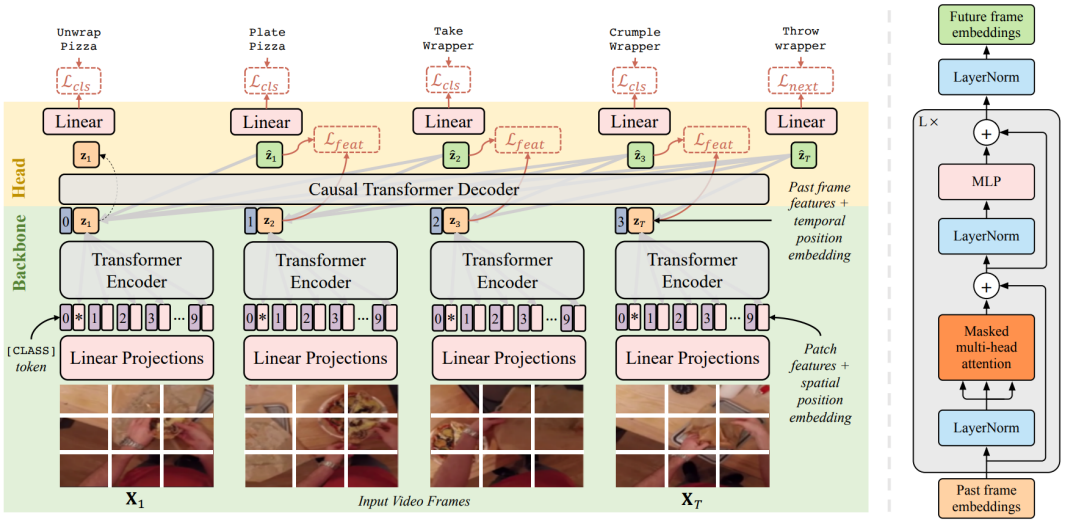

Figure 2.3: AVT diagram from [9].

Approaches based on transformer architecture such as AVT introduced in [9] have recently shown a lot of promise when applied to the task of action anticipation. The model described in [9] came top in EPIC-KITCHENS challenge in 2021. All top 3 submissions for 2022 challenge [2] were also based on this model: [11], [15] and [16], and [10]. Another example of approach based on AVT is [18].

The transformer model used in AVT allows it to process the frames in parallel. In order to ensure that only the past frames are used for prediction, AVT utilises causal masked attention, which masks video frames corresponding to the future while processing current frame. As described earlier in the dissertation, the loss function incorporates the prediction of next action, next video features and where available intermediate actions. The use of intermediate actions is meant to encourage the learning of longer-term sequences of action rather than just the one immediately following the current one.

Another difference from RU-LSTM proposed in [9] is the use of attention-based encoder in the backbone of the architecture instead of frame-based features used in [7]. This allows for a fully attention-based framework and also, with the use of the Vision Transformer proposed in [1], the use of spatial features within frames. As reported in [9], this aspect is critical to achieving performance improvement over RU-LSTM, with fully attention-based architecture outperforming RU-LSTM by around 1-2%, while AVT without using RU-LSTM features trails RU-LSTM by around 3%.

AVT head network is composed of multiple decoder layers, with the idea of focusing on increasingly high level features of the data. We explore the impact of varying the number of decoder layers on the performance of the model.

One of the most intriguing features of AVT approach is that it allows to train model to

attend to different aspects of the features simultaneously by using multiple attention heads. We explore the impact of changing the number of attention heads in our experiments.

2.3 Feature Experimentation

This project will feature an emphasis on analysing the components of the RU-LSTM architecture. This exploration will entail dissecting the primary segments of code that are pertinent in the neural network, and applying different techniques to enhance efficacy. The main goal will be to further understand the core tenets that increase accuracy within the neural network, and specifically which components boost this accuracy. The experiments will center around aspects such as loss function, optimiser and observation time as discussed in the following sections.

The success criteria for this part of the project will be the breadth of features covered and the performance improvement measured mainly by Top-5 Accuracy. Top-5 Accuracy means that any of the model's top 5 highest probability actions match the correct action. We will also occasionally use Top-1 Accuracy, which is similar but only for top probability action. When these are inconclusive, we also refer to the Time to Action (TtA)(5), which is define as the largest anticipation time (i.e., the time of earliest anticipation) in which a correct prediction has been made according to the Top-5 criterion.

2.3.1 Loss Functions

We will run experiments with changing loss function to establish if different loss functions can help yield an increase in accuracy. Specifically, we first look at the loss function of the RU-LSTM and try three loss functions from the *Pytorch* library specialised for use within action anticipation. This is followed by exploration of AVT loss function with a view to improving it and potentially combining some of its aspects with RU-LSTM.

Mean Square Error

This loss function is far simpler than the baseline cross entropy loss and is expressed as

$$L = \frac{1}{N}\sum(x_n - y_n)^2$$

where N is the number of samples, and x_n and y_n are predicted and target outputs.

Hinge

This loss function works by penalising the model based on its margin from the decision boundary. This is expressed as

$$L = \max\left(0, 1 - \hat{y}_c + \max_{i \neq c}(\hat{y}_i)\right)$$

where y_i is predictor output for class/action i and c is the target class.

Kullback-Liebler

This loss function works by computing the difference between the cross-entropy and the entropy of the results. In testing, we expected this function to perform similarly or even better than the cross-entropy loss function, due to it allowing for the entropy of probabilities to be evaluated. It is expressed as

$$L = \frac{1}{N} \sum y_n \log \left(\frac{y_n}{x_n} \right)$$

where N is the number of samples, and x_n and y_n are predicted and target outputs.

AVT loss function

The AVT loss function is comprised of three separate loss functions:

$$L = w_{next} * L_{next} + w_{feat} * L_{feat} + w_{cls} * L_{cls}.$$

The first component L_{next}, called the next-action prediction loss, is a cross-entropy loss that compares the predicted future action to the actual labeled future action. The goal of this loss is to ensure that the model accurately predicts the next action in the video.

The second component L_{feat}, called the feature-level prediction loss, leverages the causal structure of the AVT model by supervising the model's intermediate future predictions at the feature level and the action class level. Specifically, the model is trained to predict the future features that will be present in the clip and the action class that corresponds to those features. The loss function calculates the distance between the predicted features and the true future features and penalizes the model if the prediction is far from the truth.

The third component L_{cls}, called the action class level anticipative loss, leverages any action labels available in the dataset to supervise the intermediate predictions. This loss function penalizes the model if it predicts an incorrect action class for a frame that precedes the labeled action segment.

Out of these three loss functions, the RU-LSTM model already makes use of the cross entropy loss to comprise its main loss function. The third loss function I deemed to implausible to implement into the project, as I believe it is too expensive with regards to time. The second loss function is the main area that I believe could bolster the RU-LSTM model, as it can help to supervise predictions at a feature level. This would necessitate the relevant features being extracted to allow for this supervision, which may be computationally expensive. However, it would lead to higher levels of efficacy during training, with a potential combined loss function utilising cross entropy and feature-level prediction.

2.3.2 Optimiser

Optimiser is a critical component of neural network training and its choice and attributes can have decisive influence on the performance of the model. All the approaches studied in this project are based on Stochastic Gradient Descent (SGD) with different learning rate schedulers. In our experiments, We focus of testing of the impact of using different learning rate schedule on particularly in the context of AVT.

2.3.3 Observation Time

Next we hone in on manipulating the observation time within the model. Observation, also referred to as encoding, describes the video data that is taken in by the model, and describes the generic time frame in which the R-LSTM or AVTis focusing on summarising these past representations. The observation time is measured either in seconds or in frames assuming a fixed framerate, which is 4 frames per second for RU-LSTM and 1 frame per second for AVT, and represents the amount of time allowed for a model before to process video data before anticipating the next action. In [7], the authors used 6 frames (1.5s) for observation followed by 8 frames (2.0s) of anticipation, giving the total of up to 3.5s to train the model to anticipate next action. in [9] generally 10 frames are used for observation. We are interested in exploring whether training with a longer observation time will result in improved anticipation.

2.4 Hardware and Software

2.4.1 Hardware

For most of the experiments we utilised a laptop, which is an Acer Aspire 3 Laptop, equipped with the following specifications:

- **Processor:** *11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz*
- **RAM:** *16.0 GB (15.8 GB usable)*

Alongside using the laptop as the primary hardware resource for the project, we also utilised Google Cloud Platform (GCP), specifically a virtual machine instance with a single Nvidia T4 GPU. This virtual instance allows for far faster training times, around 50x quicker than using the laptop. We had limited access to this instance. We used a general purpose virtual machine to run the shorter simulation when the instance with GPU was not available as that was still faster than using the laptop. This project mixes use of both of these types of hardware, which is annotated where necessary. It should be noted that even with GCP instance equipped with a GPU, some simulations took over half a day. In general we found the available computational power to be quite limited for the experiments that we required to run, which significantly restricted our ability to complete the full scope of the project.

2.4.2 Software

RU-LSTM

This project use the software made available by the authors of [8] and [7] at [6]. The software was cloned as the baseline architecture for this project.

After downloading the RU-LSTM file, we installed anaconda and packages listed in environment.yml file of the package. This posed a number of problems as a large number of packages are not available in the version specified and some are not available at all at least on Windows. We got the installation to work after removing version restriction on packages and removing 5 packages completely. Further steps proved that they were not necessary for the system to work.

Additionally, the following changes were required to get the software to work on the laptop:

- Using *num_worker=0*
- Adding *map_location=torch.device('cpu')"* to *torch.load()* arguments

These were not used when running on the GCP instance with GPU, where they were reverted to take full advantage of the speed of the GPU offered.

AVT

AVT software has been made available from `https://github.com/facebookresearch/AVT`. It was originally downloaded to personal laptop but it was found impossible to run simulation on that due to difficulties with getting the software to run and the fact that it used a different way of accessing input data and thus required the use of the full EK-55 and EGTEA-Gaze+ datasets, which exceed the storage available on the laptop. It was therefore downloaded to the GPU-enabled instance of GCP virtual machine. The software uses Anaconda and a provides an environment definition in a similar way to RU-LSTM, which also required some tweaking to get to run on the VM.

As described above, the software is structured to allow the use of different backbone networks in combination with the AVT based head network. To get full advantage of the architecture, the AVT-b backbone is required, which uses the transformer model. It was found impossible to use that model for EK-55 due to storage requirements. The subsequent experiments on EK-55 are therefore restricted to the pre-extracted features provided with RU-LSTM, for which AVT software provides a sample configuration. We were able to use AVT-b backbone with EGTEA-Gaze+ experiments as the the size of the dataset just allowed for it to fit on our GCP when most other datasets were removed.

3 Experimental Results

All the experiments with RU-LSTM were performed on 1in10 dataset subset. The details of the dataset and network parameters for it are described in Appendix 5.2.4. For AVT, the experiments were performed on the full EK-55 dataset and EGTEA-Gaze+ dataset.

3.1 RU-LSTM Architecture Exploration

3.1.1 Sequence Completion Pre-Training

In order to test the contribution from SCP to overall performance, we compared the baseline to training without SCP but with 200 epochs of training. The overall results are presented in Figure 3.1 while full validation tables can be found in the Section 5.3.4.

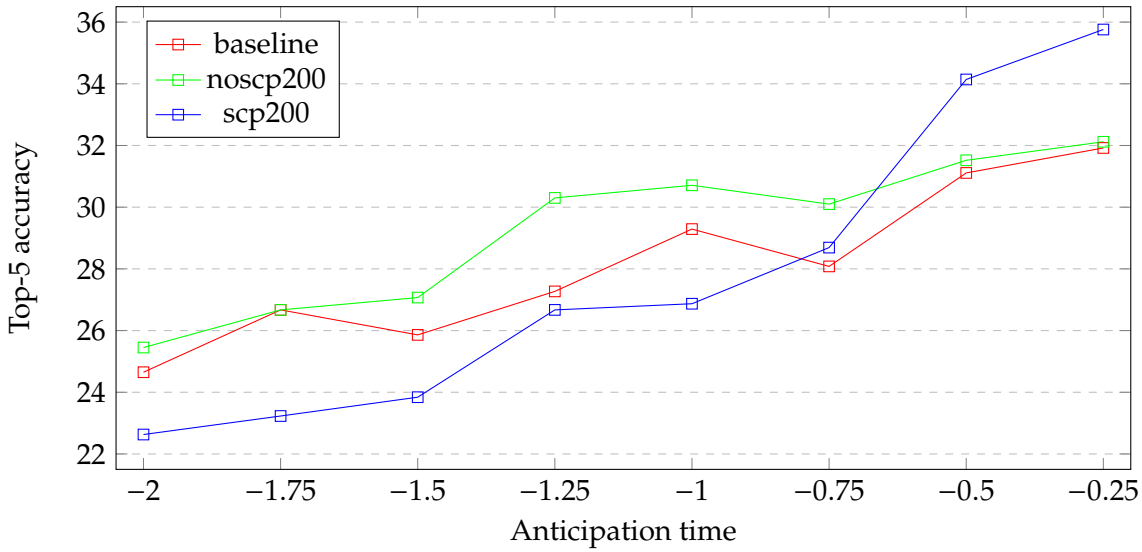

Figure 3.1: Comparison of different SCP configurations for RGB

It can be seen that for RGB modality, not using SCP actually improves performance by up to 3% at longer anticipation time, while using SCP exclusively improves performance by around 3% at very short anticipation times. This may be because SCP only provides an approximation of the actual future state, and the U-LSTM might rely too much on this approximation during pre-training. This could cause over-fitting to the training set and result in worse performance on the test set. As a result, using SCP exclusively may result in a decrease in performance at longer anticipation times, where the model needs to rely more on the R-LSTM's observation of past events.

3.1.2 Anticipating at Longer Time Scales

Experiments were run with time offset (as defined in 2.2.1) between 0.5s and 60s. The performance comparison is shown in Figure 3.2 while full validation tables can be found in Section 5.3.7.

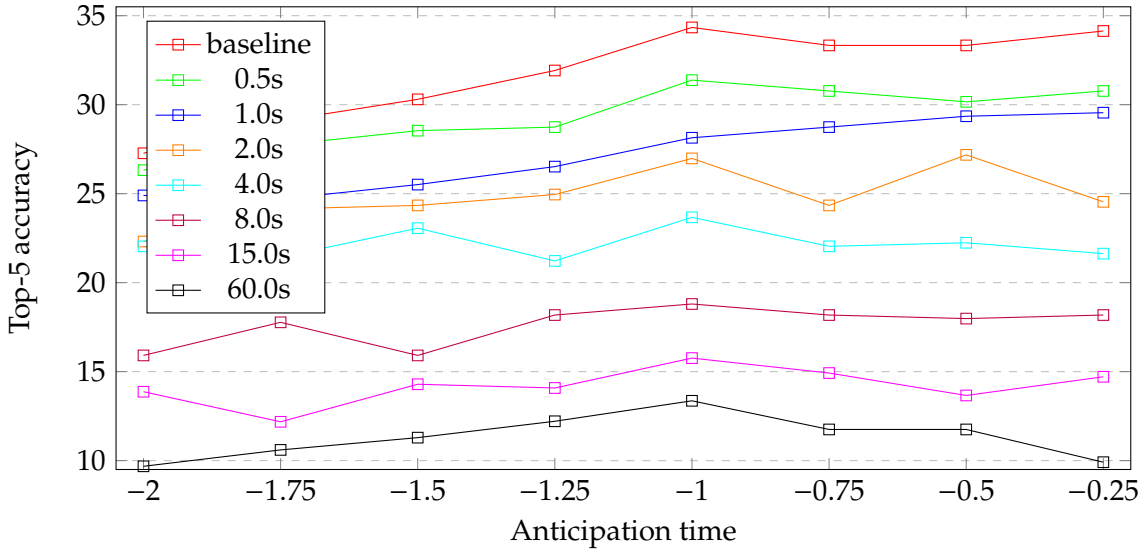

Figure 3.2: Comparison of time offsets for fusion

As could be expected, performance deteriorates as the training and anticipation times are moved further back in time. It can be seen that even at 60s the model is still capable of making some correct predictions with around 10-13% accuracy. However the performance is a lot worse compared to using immediately preceding frames. The obvious reason is that the increase of time distance from anticipated action means there is less correlation in the features between training/anticipation and action. This is compounded by the fact that the same actions take different amount of time in different clips so moving far back in time makes it increasingly more likely that the same anticipated action will be linked to a different action seen by the model.

3.2 AVT Architecture Exploration

First step in experimenting with AVT was to verify that the setup is correct by producing results matching those reported in [9]. As most of the results there are for different datasets, it was necessary to focus on results at 1s anticipation time for RGB modality only. Running the training with the default configuration, which is expected to match that used in the paper gave Top-5 accuracy of 27.78%. This matches fairly closely the result of 28.01% reported in [9].

3.2.1 Number of Layers

The baseline model used in [9] has 6 layers in AVT-h network. Here we explore the impact of varying the number of layers. The results are presented in Table 3.1.

Number of layers	Top-5 Accuracy	Top-1 Accuracy
2	27.92	12.07
6	28.38	12.57
12	27.78	12.54

Table 3.1: Results with different number of layers

It is seen that the size of the network is indeed roughly optimal for the task on this dataset and increasing it actually hurts performance. This may be because the model becomes too complex resulting in over-training.

3.2.2 Attention Heads

Here we explore the impact of varying the number of future attention heads from the baseline used in [9]. The results are presented in Table 3.2.

Number of heads	EK-55 Top-5 Accuracy	EGTEA-Gaze+ Top-5 Accuracy
1		71.22
2	28.16	71.17
4	27.98	71.51
8	27.78	72.06
16	27.68	

Table 3.2: Results with different number of attention heads

Somewhat surprisingly, best performance for EK-55 was obtained with 2 attention heads, implying that the baseline value of 8 is unnecessarily high. This result suggests that there is no benefit in allowing the model to use a large number of temporal patterns for prediction. This may be because the temporal dimension is less feature rich than spatial one, where there are typically a large number of objects in the scene. In contrast, the temporal patterns for EK-55 appear to be fairly repetitive thus favouring a small number of attention heads.

For EGTEA-Gaze+, increasing the number of attention heads does improve performance. This is surprising as only 3 observation frames were used in this experiment and suggests that the actions in EGTEA Gaze+ have more complex temporal content for additional attention heads to take advantage of.

3.3 Feature Experimentation

3.3.1 Loss Function

The results of experiments with loss functions are presented in Figure 3.3 while full validation tables can be found in Appendix 5.3.3.

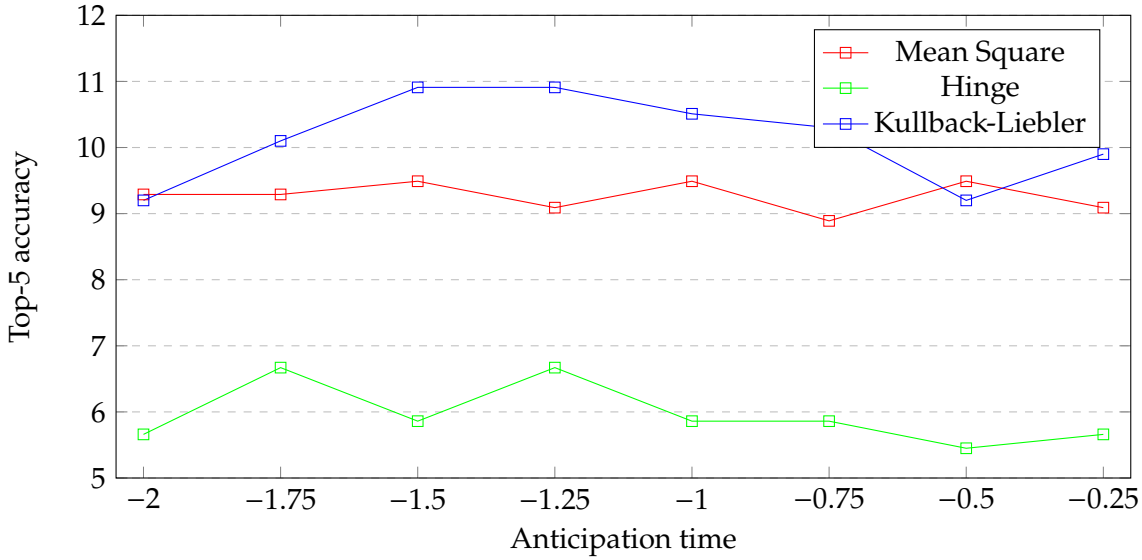

Figure 3.3: Comparison of loss functions.

After examining the results and the graphs of the loss functions, we ascertained a few conclusions. The **Hinge loss** performed very poorly, which we believe is a result of hinge loss harshly punishing incorrect classifications, as well as correct but non-confident classifications. This may have led to minor errors in classification causing larger error margins. The **Kullback-Liebler** function performed comparatively well at earlier anticipation times, however similarly showed high variance. This could be a result of the log function leading to vast magnitude when components of the function are very small. **Mean square error** was generally more inaccurate, but more consistent, with the highest value at 0.5s anticipation time. We believe that this is because the loss function uses an average, leading to more reproducible results. One issue we found was that it far amplifies error margins, due to squaring the errors. This led to outliers skewing the predictions, giving low accuracy in the training and validation.

AVT Loss Function

Second experiment was performed to explore the loss function of AVT and in particular the impact of varying the weight of feature prediction (L_{feat}) from the baseline of [9], where 1.0 was used for both L_{next} and L_{feat} and 0.1 for L_{cls}. We expected that increasing the weighting would strengthen the self-supervision aspect of training while lowering it would result in training more directly targeting anticipation accuracy. EGTEA-Gaze+ dataset with 6 observation frames were used in this experiment to allow the temporal dimension to be exercised as much as computational resources allowed. Figure 3.4 illustrates the evolution of Top-5 Accuracy during training for different values of the feature weight. For completeness, the Top-1 Accuracy for 0.5, 1.0, and 2.0 weights were 43.72, 44.26 and 42.98 respectively.

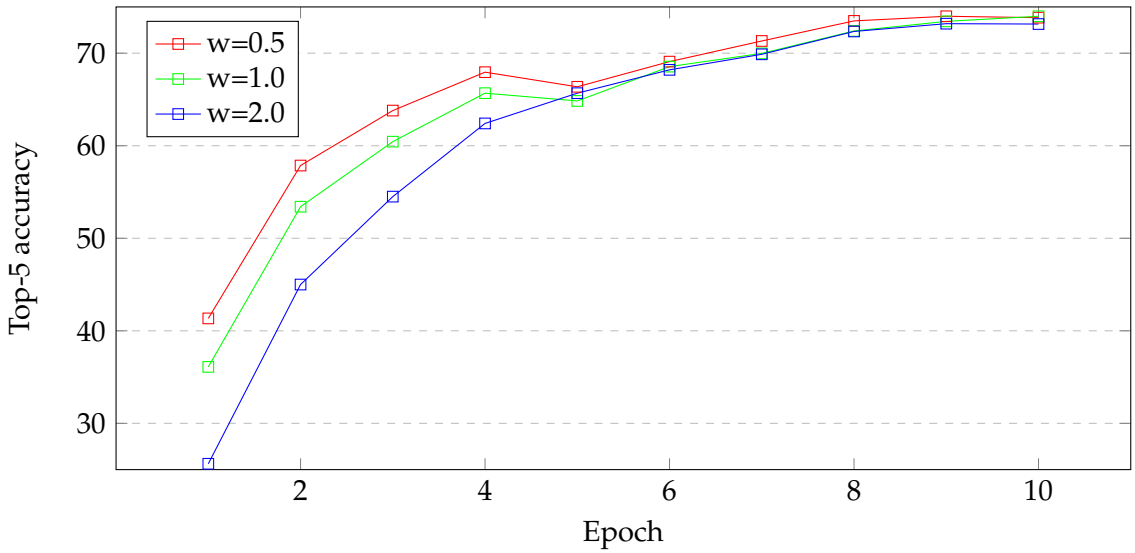

Figure 3.4: Top-5 Accuracy change during training for different weights

It is seen that lowering the weight of the feature cost improves the results somewhat. This suggests that at least with the limited observation and anticipation time, the self-supervision does not result in improved overall model. It can also be seen that the accuracy for different weights starts fairly far apart and converges to fairly close values. This may be an indication that increasing the weight may indeed result in a more coherent model.

3.3.2 Optimiser

In all the experiments we followed the optimizer settings from [9], which for EGTEA-Gaze+ specify SGD with cosine scheduler and 5 warm-up epochs with linear learning rate increase. Figure 3.4 suggests that with 5 warm-up epochs the learning rate may still go up too fast resulting in early over-fitting. In order to explore the impact of the optimiser schedule on training behaviour and performance of the final model, we next modify the configuration of the scheduler by varying the warm up period between 3-7 epochs. The results for 3 observation frames are presented in Table 3.3.

Warmup epochs	Top-5 Accuracy	Top-1 Accuracy
3	72.02	42.78
5	71.51	43.03
7	72.60	43.08

Table 3.3: Results for EGTEA-Gaze+ with different number of attention heads

As expected increasing the number of epochs of warm-up leads to slower initial improvement. It avoids the dip in performance around fifth epoch but there is still a flattening in the rate of accuracy improvement. A possible explanation is that the model goes through a fairly flat area of loss function landscape and while keeping learning rate lower allows to avoid the dip, using higher rate allows it to move past it quicker as we can see from improved accuracy in sixth epoch. Another aspect is sharp decline in improvement in later iterations, which may be caused by too quick annealing with the cosine scheduler with a small number of epochs (3 in this case). The rate may go down too quickly and it would be possible to more accurately locate the final minimum of loss function with more epochs and slower learning towards the end.

3.3.3 Observation Time

In this section we explore the impact of changing the observation time on the performance of RU-LSTM. We use only RGB modality and the 1in10 subset of the dataset for this experiment. As this experiment required the use of video frames beyond the main dataset used in [7], we downloaded the full set of extracted features, which amounts to around 100GB of storage.

We ran the model following the methodology of [7], i.e. an SCP pre-training with 100 epochs followed by 100 epochs of training. In addition to the original observation time of 6 frames, we tried 4, 8, 16 and 24. The overall results for Top-5 Accuracy are presented in Figure 3.5 while detailed results for all classes and metrics can be found in Appendix 5.3.5.

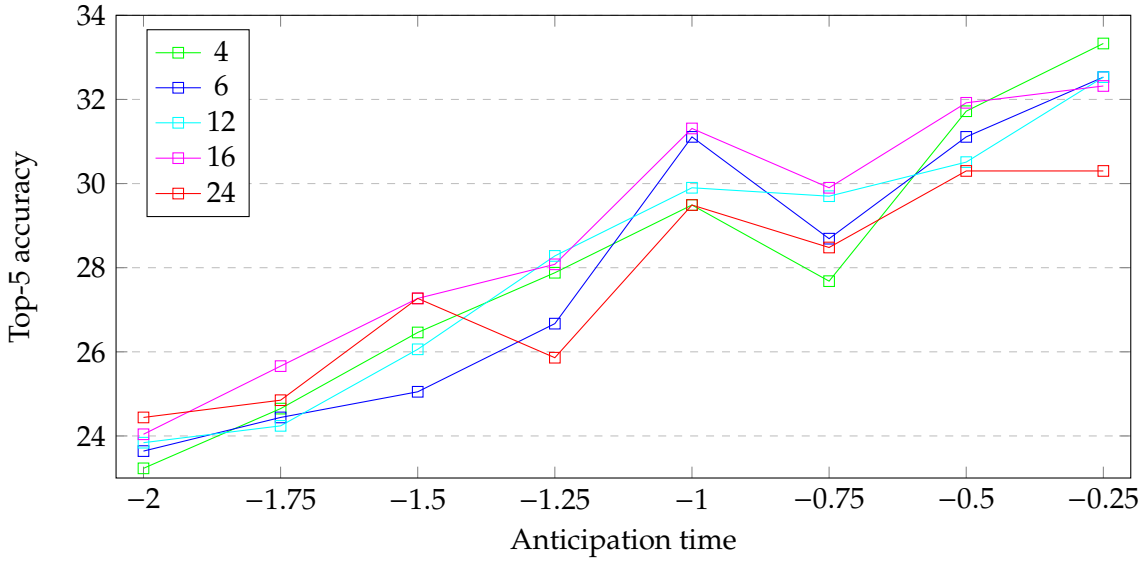

Figure 3.5: Top-5 Accuracy for RGB modality for different numbers of observation frames

The results are quite noisy and in fact for some anticipation times the best results are achieved for short observation times. Part of the reason is probably the general noisiness of the results for the 1in10 subset of the data. Another reason is probably that the actions take different amounts of time, so adding more observation time sometimes gives more context for a single action and sometimes introduces a new action which may tend to confuse the model.

The TtA(5) results further corroborate the conclusion that there is no clear improvement as they have the highest value (0.73) for 2 observation frames, with seemingly random variations between 0.69–0.73 and ending up at 0.70 for 24 frames. In general it seems that there is little to gain by going beyond the default observation time of 6 frames. This matches the conclusion in the ablation studies in [7].

AVT

We performed similar experiments with AVT with EK-55 dataset. It was impossible to use more than 16 frames on our hardware. The results are presented in Table 3.4.

observation frames	Top-5 Accuracy	Best iteration
3	27.76	25
6	27.80	27
10	27.78	20
16	27.48	23

Table 3.4: Results with different number of frames

It can be seen that the performance does not improve significantly with the increase in the number of frames used for training. This is in agreement with the results for RU-LSTM.

Another observation that was found of interest is that the training for AVT converges quite quickly, typically in under 25 epochs, which is significantly quicker than is the case for RU-LSTM, where typically the number of epochs approaching 100 is required for both SCP and training stages.

4 Project Conclusion

4.1 Discussion of Experimental Results

In general, exploration and experiments we were able to perform were severely constrained by computational resources at our disposal.

After experimenting with the loss functions contained in the *Pytorch* library, we concluded that there was no tangible performance boost from any of them. This led us to further inquire into the loss function utilised in the AVT architecture [9]. The varying of weights in AVT showed more promise, with larger weight for feature loss leading to smoother accuracy increase in training. A potential area for further research would be varying the relative weights of loss function components to initially focus more on predicting video features, and then moving on to strengthening the supervised anticipation component. It would also be interesting to see whether such an approach would benefit longer term anticipation.

For the optimiser function, we obtained some interesting results for scheduling of learning rate, which illustrate the trade-offs of exploring the loss function landscape. With a small number of epochs for warm-up and then cosine annealing, the training suffers from both early over-training and then low accuracy in locating the minimum of loss function. Generally, our limited computational resources restricted us to using a small number of training epochs, and it would be very interesting to be able to run longer experiments to observe the full impact of the tuning of the schedule.

After exploring how observation time affects model performance, we garnered a few conclusions. At longer anticipation times, the increased amount of observation frames produced higher accuracy metrics. However, at shorter anticipation times, the lower amount of observation frames gave better results. Thus an adaptive system with models trained on different number of frames used for different anticipation times could be more performant though expensive to implement. A more promising direction might be to use a fixed number of previous actions for training rather than a fixed number of frames, which would address the problem of different samples containing different number of whole and partial actions.

The conclusion of the experiments with RU-LSTM sequence completion pre-training is that the use of SCP does not give a significant performance boost overall although it seems beneficial for very short anticipation times. The improvements reported in [7] may have simply been the result of the higher overall number of epochs resulting from adding pre-training step.

For longer term anticipation, we observed expected deterioration with increasing passage of time. Better performance could likely be achieved if the anticipation was performed on the basis of moving by a certain number of actions rather than seconds. Another interesting direction of further research on RU-LSTM could be unrolling the model by a frame after each anticipation and comparing behaviour to that explored in [9].

AVT with pre-extracted TSN features achieves around 28% Top-5 Accuracy on EK-55 dataset. This is quite significantly below 31% obtained for RGB modality with RU-LSTM and highlights that the decoder part of the network on its is not sufficient to provide improvement over that architecture.

The experiments with the number of attention heads indicate that their optimal number depends on the dataset. It suggests that it would be interesting to investigate an adaptive approach where the number of attention heads would be determined based on analysis of the salient features of the video data. An example of such a feature could be a measure of the magnitude of optical flow.

4.2 Conclusion

We confirmed the performance results of RU-LSTM approach and explored some of its salient aspects, concluding that in particular SCP may be of limited benefit. We investigated the potential improvements from modifying certain aspects of the model as well as general network features and architecture for both RU-LSTM and AVT and obtained some promising results.

We confirmed performance results reported in [9] on the same features used in [7]. We noted that they are worse than the performance reported in [7] using RU-LSTM. We therefore conclude that the backbone of the AVT architecture is critical to obtain the performance reported in the paper. In further work, it would be interesting to explore the whether the combination of AVT backbone and RU-LSTM temporal component could improve performance as the improvement in AVT seems to come from backbone.

The exploration was limited by lack of computational resources, and it would thus be extremely interesting to continue the research on combining the features of the two studied approaches, as well as the more specific potential improvements outlined in Section 4.1.

5 Appendix

5.1 Discarded Ideas

5.1.1 RU-LSTM

Loss with Multiple Correct Answers

I thought that the loss function with multiple “correct” answers could be considered due to the fact that anticipation is an ill-defined problem. This was partially guided by the fact that the performance results are frequently reported for "top-5" accuracy. However, on closer examination I realised that it is not possible because we do not have top-5 correct labels; instead we take top-5 predictions and say we are correct if one of these 5 is correct. Thus ground truth is only available for a single "correct" action.

Using Action Frames for Pre-Training

This was the idea to extend SCP to not just look at future "anticipation" actions for training but also frames from "the future" belonging to the action to be anticipated. While it seems that it would be possible to implement, after some thought I decided that it may not be helpful as it creates confusion between the previous actions that are the basis for anticipation and the action to be anticipated, which may hinder rather than help learning to anticipate.

5.1.2 AVT

Explore the backbone network. This turned out to be impossible without moving away completely from the baseline and re-training the backbone network from scratch as [9] and software used pre-trained Visual Transcoder data for frame-level features.

5.2 Further Details on Datasets

5.2.1 CSV Files

- *actions.csv*: This file contains a list of all the action classes in the dataset, along with their associated verb classes and noun classes. It yields all the possible actions in the dataset.
- *epic_many_shot_nouns.csv*: This file contains a list of the object classes in the dataset, along with the number of times they appear in the training set.
- *test_seen.csv*: This file contains the annotations for the *seen* test set, which includes videos from the same participants as the training set. This set is used to evaluate how well the model can generalize actions from the same participants.
- *test_unseen.csv*: This file contains the annotations for the *unseen* test set, which includes videos from participants that were not included in the training set. This set is useful to see how the model can generalise actions from different participants.
- *training_videos.csv*: This file contains a list of all the videos in the training set, along with metadata such as the participant ID, the video ID, the duration, and the resolution.
- *training.csv*: This file contains the annotations for the training set, including information about the action segments, object instances, and interactions between people and objects.
- *validation_videos.csv*: This file contains a list of all the videos in the validation set, along with metadata such as the participant ID, the video ID, the duration, and the resolution.
- *validation.csv*: This file contains the annotations for the validation set, which is used for fine-tuning models and selecting hyper-parameters.

5.2.2 Extracting a Subset of the Dataset

The full EPIC-KITCHENS-55 dataset (which will henceforth be referred to as EK55) takes a very long time to process on the main computing resource (personal laptop without dedicated GPU). The short simulation we ran suggested that it would take 50 hours to run a single modality training resulting in 350 hours for full training including modality fusion). In order to make it practical to conduct experiments, we decided to create a more manageable subset.

Extracting a Given Percentage of the Subset

After some simple experiments, I realised that in order to obtain sensible results and be consistent with the general methodology, I needed to first identify a subset of actions to be present in both training and validation sets and then extract video fragments from both that contain only this actions. I then proceeded to implement a python script that first identifies a fixed percentage of action in validation set, then selects video segments labeled with these actions up to the same percentage of the total number of video segments in original validation set, and finally performs the same process on the training set using pre-selected actions.

5.2.3 Discarded Clips with Long Anticipation Time

The number of discarded sequences varied between 1 and 342, as illustrated in Table 5.1. For 60s time offset, it represents around 15% of the total number of available sequences. While this is a significant percentage it should not invalidated overall conclusions as the trends observed in the results are quite clear.

Time offset	Discarded training	Discarded validation
0.0s	1	2
0.5s	7	3
1.0s	9	3
2.0s	17	4
4.0s	39	7
8.0s	75	13
15.0s	114	21
30.0s	188	38
60.0s	342	63

Table 5.1: Number of discarded sequences per time offset

5.2.4 Tuning of RU-LSTM Parameters for 1in10 Subset

After some initial experiments, we decided to focus further experiments on a subset representing 10% of the original dataset, which contained 234 actions, training set of size 2,349 and validation set of size 497. For this dataset, a single modality training run of 100 epochs with hidden layer of size 1024 takes around 4hr on my laptop for RGB and flow, and around 3 hours for object modality. Initial experiments on this subset showed that the model performed very well on training data but perhaps did not generalise very well to validation set. It also still took quite a long time to run full training on all modalities and fusion. In order to find a better balance between performance, risk of over-training and training time we therefore decided to experiment with reducing the size of the hidden layers in LSTM networks.

In order to obtain the results in reasonable time, we restricted the experiments with the size of the hidden layer to RGB modality. The corresponding tables presenting full results for hidden layer sizes between 128-1024 can be found in Section 5.3.2. The overall results are presented in Figure 5.1.

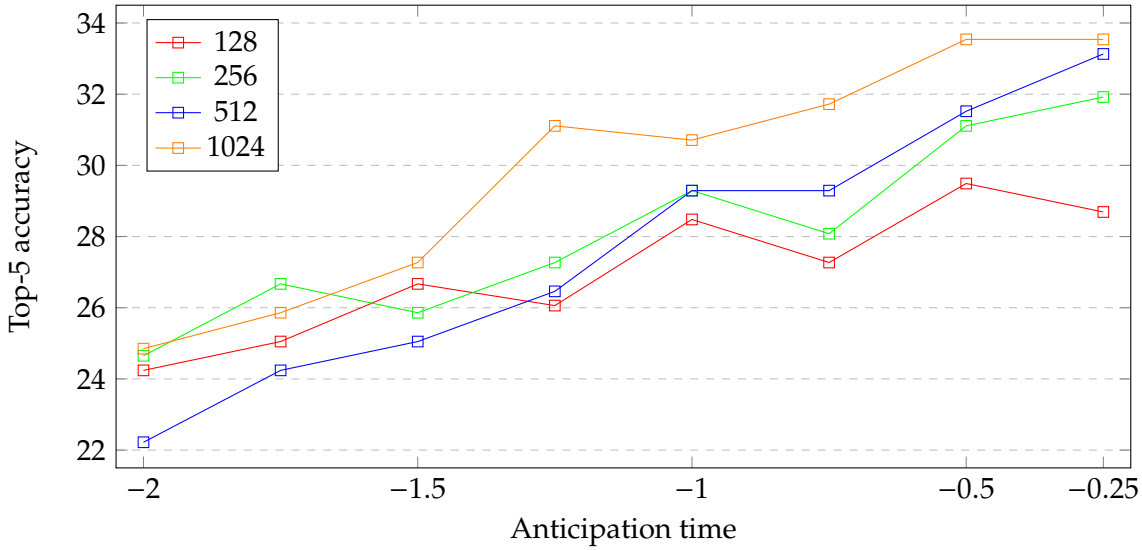

Figure 5.1: Performance comparison for different sizes of hidden layer

It can be seen that in general the Top-5 accuracy improves with the increase in the size of the hidden layer. There are some significant deviations from this general trend, notably for hidden size of 512 at long anticipation times, where it performs worse than 128 and 256 cells. While hidden layer of size 1024 performs the best overall, hidden layer of size 256 generally remains within around 2% of that and is generally quite consistent across the tested range of anticipation times.

The larger hidden layers appear to exhibit some symptoms of over-training, with e.g. training accuracy of 92% for 512 and 98% for 1024 cells in the hidden layer versus validation accuracy around 29% in both cases at the end of the training run, suggesting that the network may be too big for the size of the training set and the number of classes. Given also that the computation time roughly doubles with doubling of the number of cells in hidden layer, it was concluded that 256 appears to be a good compromise between raw anticipation results on the one hand and training time on the other, particularly as results with high number of hidden cells seem to indicate over-training on the 1in10 training set. Henceforth, unless noted otherwise, the results for RU-LSTM will be reported for this 1in10 subset and model with 256 cells in the hidden layer.

For completeness, the remaining modality tables for hidden layer with 256 cells can be found in Section 5.3.6.

5.3 Detailed Experimental Results

5.3.1 Results for Full EK-55 Dataset

The project utilised intermittent access to a GCP machine with a single GPU. This had sufficient computing power to run experiments on the full dataset. The training for one modality took a little over 1 hour compared to around 40 hours on the laptop used. Unfortunately it was not always possible to start it once it was stopped so we only have a limited set of results. These

tables can be found at 5.3.8. It can be seen that the results are comparable to those reported in [7], e.g. Top-5 accuracy for action anticipation is 38.54 versus 38.98 reported in [7].

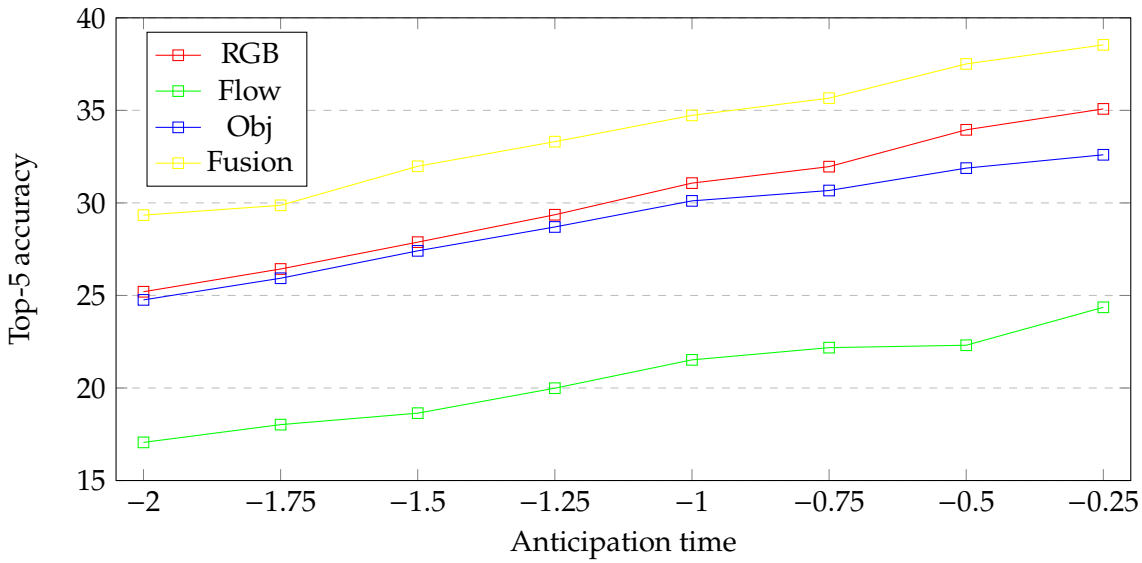

Figure 5.2: Full EK-55 modality results

5.3.2 Validation Tables: Hidden Layer

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	32.73	32.73	34.55	31.72	34.34	36.97	35.56	35.15
	Top-5 Accuracy	72.93	74.75	73.33	73.54	73.33	73.33	74.14	74.75
	Mean Top-5 Recall	26.99	29.17	28.04	27.34	27.42	27.77	28.60	27.79
Noun	Top-1 Accuracy	15.56	16.97	15.76	15.96	17.17	16.77	18.59	17.98
	Top-5 Accuracy	34.34	35.35	34.14	35.96	36.97	38.79	42.22	40.20
	Mean Top-5 Recall	25.35	25.99	24.46	24.94	27.35	28.52	30.78	28.77
Action	Top-1 Accuracy	09.29	09.29	10.51	09.70	11.72	10.51	11.92	12.32
	Top-5 Accuracy	24.24	25.05	26.67	26.06	28.48	27.27	29.49	28.69
	Mean Top-5 Recall	13.61	14.18	14.92	14.46	15.94	14.44	15.44	14.29
Mean TtA(5): VERB: 1.54 NOUN: 0.89 ACTION: 0.69									

Table 5.2: Validation results for hidden layer with 128 cells, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.49	29.09	33.13	31.52	33.94	35.35	35.96	34.75
	Top-5 Accuracy	71.52	71.31	71.92	72.12	71.11	71.72	72.32	71.52
	Mean Top-5 Recall	27.23	27.26	28.09	28.05	29.10	28.69	28.88	28.56
Noun	Top-1 Accuracy	17.98	16.97	17.37	17.58	19.39	19.19	21.62	21.01
	Top-5 Accuracy	35.76	36.77	35.76	38.79	37.98	40.00	40.61	41.62
	Mean Top-5 Recall	29.88	30.14	28.72	29.82	30.12	29.80	30.65	32.42
Action	Top-1 Accuracy	10.51	10.10	11.31	12.12	13.13	14.14	14.55	13.74
	Top-5 Accuracy	24.65	26.67	25.86	27.27	29.29	28.08	31.11	31.92
	Mean Top-5 Recall	13.55	15.35	14.72	14.62	17.20	14.90	16.79	16.96
Mean TtA(5): VERB: 1.54 NOUN: 0.92 ACTION: 0.71									

Table 5.3: Validation results for hidden layer with 256 cells, RGB modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.48	26.67	29.70	29.09	30.10	33.33	34.55	32.93
	Top-5 Accuracy	67.68	68.28	69.09	68.89	66.87	67.68	68.28	67.47
	Mean Top-5 Recall	26.41	28.51	28.01	27.96	29.56	29.48	30.03	29.70
Noun	Top-1 Accuracy	16.77	15.96	16.36	15.76	20.20	19.39	21.01	22.63
	Top-5 Accuracy	34.55	36.57	35.35	39.39	39.80	41.41	41.41	43.03
	Mean Top-5 Recall	29.26	29.78	27.61	33.20	32.87	31.63	34.53	34.60
Action	Top-1 Accuracy	08.89	09.09	09.90	09.90	13.13	13.54	14.55	14.55
	Top-5 Accuracy	22.22	24.24	25.05	26.46	29.29	29.29	31.52	33.13
	Mean Top-5 Recall	14.04	14.76	15.39	15.63	18.24	17.62	18.18	19.77
Mean TtA(5): VERB: 1.52 NOUN: 0.94 ACTION: 0.70									

Table 5.4: Validation results for hidden layer with 512 cells, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.49	29.09	31.52	30.91	30.71	33.13	32.93	34.14
	Top-5 Accuracy	67.68	67.47	70.10	68.89	67.68	66.67	66.26	66.87
	Mean Top-5 Recall	30.00	28.50	31.74	32.40	30.94	30.72	31.55	30.92
Noun	Top-1 Accuracy	15.35	15.76	16.57	18.79	19.19	20.00	21.82	22.63
	Top-5 Accuracy	35.56	36.36	38.18	39.80	41.01	42.22	43.43	45.25
	Mean Top-5 Recall	32.66	32.09	31.35	34.67	34.37	36.96	38.31	40.05
Action	Top-1 Accuracy	09.70	10.30	11.52	11.92	13.74	13.33	15.15	15.76
	Top-5 Accuracy	24.85	25.86	27.27	31.11	30.71	31.72	33.54	33.54
	Mean Top-5 Recall	16.09	15.18	15.41	17.92	18.13	19.03	21.44	19.64
Mean TtA(5): VERB: 1.50 NOUN: 0.96 ACTION: 0.74									

Table 5.5: Validation results for hidden layer with 1024 cells, RGB

5.3.3 Validation Tables: Loss Function

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
	Top-5 Accuracy	71.11	71.11	71.11	71.11	71.11	71.11	71.11	71.11
	Mean Top-5 Recall	19.23	19.23	19.23	19.23	19.23	19.23	19.23	19.23
Noun	Top-1 Accuracy	04.44	04.44	04.44	04.44	04.44	04.44	04.44	04.44
	Top-5 Accuracy	15.15	15.15	15.15	15.15	15.15	15.15	15.15	15.15
	Mean Top-5 Recall	07.58	07.58	07.58	07.58	07.58	07.58	07.58	07.58
Action	Top-1 Accuracy	02.83	02.63	02.63	02.63	02.63	02.22	02.42	02.63
	Top-5 Accuracy	09.29	09.29	09.49	09.09	09.49	08.89	09.49	09.09
	Mean Top-5 Recall	02.17	02.17	02.23	02.13	02.23	02.04	02.18	02.15
Mean TtA(5): VERB: 1.42 NOUN: 0.30 ACTION: 0.19									

Table 5.6: Results for Mean Square Error Loss

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
	Top-5 Accuracy	71.11	71.11	71.11	71.11	71.11	71.11	71.11	71.11
	Mean Top-5 Recall	19.23	19.23	19.23	19.23	19.23	19.23	19.23	19.23
Noun	Top-1 Accuracy	04.44	04.44	04.44	04.44	04.44	04.44	04.44	04.44
	Top-5 Accuracy	14.75	14.34	15.15	14.95	15.15	14.75	14.95	14.95
	Mean Top-5 Recall	07.42	07.27	07.58	07.50	07.58	07.42	07.54	07.54
Action	Top-1 Accuracy	01.41	01.01	01.82	02.22	01.41	01.62	01.21	00.81
	Top-5 Accuracy	05.66	06.67	05.86	06.67	05.86	05.86	05.45	05.66
	Mean Top-5 Recall	02.09	02.23	02.08	03.45	02.24	02.33	01.77	01.85
Mean TtA(5): VERB: 1.42 NOUN: 0.30 ACTION: 0.26									

Table 5.7: Results for Hinge Loss

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	20.61	18.99	21.62	21.01	22.22	22.02	23.43	20.61
	Top-5 Accuracy	71.11	71.11	71.11	71.11	71.11	71.11	71.11	71.11
	Mean Top-5 Recall	19.23	19.23	19.23	19.23	19.23	19.23	19.23	19.23
Noun	Top-1 Accuracy	05.25	04.85	04.65	04.24	04.24	05.86	04.24	05.25
	Top-5 Accuracy	21.41	21.41	22.22	22.02	22.02	21.21	21.82	22.22
	Mean Top-5 Recall	07.94	08.02	08.30	08.27	08.27	07.95	08.15	08.26
Action	Top-1 Accuracy	01.41	02.83	03.43	02.63	03.23	02.83	02.63	02.83
	Top-5 Accuracy	09.70	10.10	10.91	10.91	10.51	10.30	09.70	09.90
	Mean Top-5 Recall	02.38	02.50	02.68	02.80	02.59	02.53	02.35	02.43
Mean TtA(5): VERB: 1.42 NOUN: 0.47 ACTION: 0.26									

Table 5.8: Results for Kullback-Liebler Loss

5.3.4 Validation Tables: SCP

No SCP

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.89	28.08	31.72	31.11	33.13	33.54	35.35	34.75
	Top-5 Accuracy	71.72	71.31	72.32	73.33	72.93	71.31	73.13	73.13
	Mean Top-5 Recall	27.79	27.46	30.50	30.21	31.15	29.28	30.41	29.23
Noun	Top-1 Accuracy	14.95	15.15	15.15	17.37	18.38	18.79	21.41	20.61
	Top-5 Accuracy	36.57	37.17	38.38	41.01	41.82	41.41	43.64	44.04
	Mean Top-5 Recall	28.39	29.04	28.63	32.97	31.97	31.39	34.24	35.62
Action	Top-1 Accuracy	09.09	08.28	10.71	10.71	13.54	12.53	14.34	13.94
	Top-5 Accuracy	25.45	26.67	27.07	30.30	30.71	30.10	31.52	32.12
	Mean Top-5 Recall	15.33	15.53	15.34	17.06	17.76	17.40	18.37	18.32
Mean TtA(5): VERB: 1.54 NOUN: 0.95 ACTION: 0.74									

Table 5.9: Validation results for no SCP with 200 epochs, RGB modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	27.88	29.09	29.49	31.11	30.30	32.12	33.74	31.11
	Top-5 Accuracy	73.74	71.92	73.33	73.54	73.54	73.74	74.95	74.34
	Mean Top-5 Recall	24.66	23.04	24.44	24.51	23.87	25.97	27.01	25.55
Noun	Top-1 Accuracy	09.29	07.88	09.29	09.29	09.70	11.72	13.33	11.92
	Top-5 Accuracy	25.45	25.66	25.05	26.67	25.25	26.67	27.47	29.90
	Mean Top-5 Recall	14.03	13.57	12.78	13.33	12.85	13.99	13.89	15.45
Action	Top-1 Accuracy	06.46	05.05	05.66	06.87	05.86	06.67	08.28	07.07
	Top-5 Accuracy	17.78	17.37	17.37	18.59	18.59	19.39	19.60	20.20
	Mean Top-5 Recall	07.14	05.98	06.27	06.60	07.00	07.59	07.66	08.05
Mean TtA(5): VERB: 1.53 NOUN: 0.70 ACTION: 0.51									

Table 5.10: Validation results for no SCP with 200 epochs, flow modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.29	28.69	28.28	29.09	29.90	29.70	30.10	28.69
	Top-5 Accuracy	74.55	73.54	73.33	73.94	75.15	75.35	74.14	72.12
	Mean Top-5 Recall	28.87	27.48	27.80	28.33	29.16	29.25	28.08	24.14
Noun	Top-1 Accuracy	16.97	16.16	17.37	17.17	18.79	19.19	19.39	18.79
	Top-5 Accuracy	38.38	37.17	37.78	37.17	38.79	42.22	42.22	38.99
	Mean Top-5 Recall	31.09	29.42	29.86	29.27	30.20	32.11	30.59	23.92
Action	Top-1 Accuracy	08.69	09.09	08.48	08.08	09.49	10.10	10.30	10.10
	Top-5 Accuracy	24.04	24.04	26.46	26.46	28.08	29.09	28.48	26.87
	Mean Top-5 Recall	14.38	13.63	14.97	14.98	15.73	15.40	14.14	12.74
Mean TtA(5): VERB: 1.54 NOUN: 0.91 ACTION: 0.63									

Table 5.11: Validation results for no SCP with 200 epochs, obj modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	31.31	32.12	33.94	33.33	34.55	36.36	36.16	37.17
	Top-5 Accuracy	70.91	71.11	73.94	73.33	72.93	73.74	72.73	72.12
	Mean Top-5 Recall	29.99	29.40	31.75	31.06	32.08	32.48	32.81	30.41
Noun	Top-1 Accuracy	16.97	16.97	18.99	19.60	20.40	21.41	22.42	21.41
	Top-5 Accuracy	37.78	40.40	42.02	43.23	46.46	45.66	46.46	46.67
	Mean Top-5 Recall	31.15	32.37	34.46	36.74	40.67	37.03	40.70	38.70
Action	Top-1 Accuracy	10.71	10.30	12.32	11.92	13.74	13.94	15.35	14.55
	Top-5 Accuracy	27.07	27.47	30.30	32.93	34.34	34.34	34.14	33.54
	Mean Top-5 Recall	16.72	17.25	18.17	19.71	20.61	20.80	20.04	19.71
Mean TtA(5): VERB: 1.55 NOUN: 0.98 ACTION: 0.75									

Table 5.12: Validation results for no SCP with 200 epochs, fusion

SCP WITH 200 EPOCHS

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.29	27.68	29.90	30.91	32.12	34.55	37.17	36.57
	Top-5 Accuracy	70.51	67.27	70.51	70.71	69.90	70.91	71.72	71.31
	Mean Top-5 Recall	29.74	25.85	29.27	27.82	28.62	30.45	30.31	29.09
Noun	Top-1 Accuracy	16.77	16.57	16.97	18.38	19.80	19.60	21.82	24.65
	Top-5 Accuracy	35.35	34.55	35.35	39.80	38.79	42.02	45.66	46.06
	Mean Top-5 Recall	30.32	29.00	28.13	32.25	31.89	33.68	38.87	37.50
Action	Top-1 Accuracy	09.29	09.70	10.51	10.51	12.53	13.33	15.35	15.96
	Top-5 Accuracy	22.63	23.23	23.84	26.67	26.87	28.69	34.14	35.76
	Mean Top-5 Recall	14.14	14.63	14.01	15.43	16.01	17.05	20.60	21.23
Mean TtA(5): VERB: 1.57 NOUN: 0.95 ACTION: 0.71									

Table 5.13: Validation results for SCP with 200 epochs, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	24.24	25.66	27.68	25.25	25.66	29.70	34.55	35.15
	Top-5 Accuracy	69.90	67.27	69.09	69.49	69.90	70.71	72.73	72.73
	Mean Top-5 Recall	26.39	23.26	24.22	24.84	25.16	28.01	27.21	27.43
Noun	Top-1 Accuracy	09.29	07.07	08.08	07.27	10.71	10.71	12.32	12.53
	Top-5 Accuracy	23.23	23.84	25.45	26.26	25.05	26.87	28.89	30.71
	Mean Top-5 Recall	13.17	13.55	14.77	14.95	14.75	15.57	16.73	17.67
Action	Top-1 Accuracy	04.65	03.03	05.45	04.44	05.86	07.47	07.88	07.68
	Top-5 Accuracy	14.55	14.14	15.15	16.77	17.78	18.18	21.21	21.01
	Mean Top-5 Recall	06.88	06.25	05.62	06.86	07.36	08.34	09.20	08.87
Mean TtA(5): VERB: 1.54 NOUN: 0.78 ACTION: 0.55									

Table 5.14: Validation results for SCP with 200 epochs, flow

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.70	30.10	31.92	30.10	29.29	31.72	32.12	32.32
	Top-5 Accuracy	74.14	74.14	74.34	74.95	76.16	76.16	75.76	74.95
	Mean Top-5 Recall	28.13	27.81	28.56	28.58	29.83	29.85	29.82	27.88
Noun	Top-1 Accuracy	15.76	16.97	17.98	18.38	18.79	20.00	20.61	20.00
	Top-5 Accuracy	36.57	35.15	38.18	37.58	40.40	42.42	43.23	41.62
	Mean Top-5 Recall	30.52	28.85	30.85	29.48	32.24	34.64	33.96	30.15
Action	Top-1 Accuracy	08.48	09.70	10.10	09.70	11.52	11.72	11.92	11.72
	Top-5 Accuracy	22.42	23.43	25.05	25.45	26.67	28.48	26.87	28.28
	Mean Top-5 Recall	13.59	14.26	14.59	14.25	14.99	15.67	14.60	15.13
Mean TtA(5): VERB: 1.53 NOUN: 0.89 ACTION: 0.61									

Table 5.15: Validation results for SCP with 200 epochs, obj

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	32.12	31.72	33.33	34.14	34.75	36.36	37.58	36.16
	Top-5 Accuracy	73.94	72.73	74.95	73.94	73.94	74.34	74.55	74.55
	Mean Top-5 Recall	31.44	27.86	30.45	30.14	30.42	29.60	31.36	30.93
Noun	Top-1 Accuracy	17.37	17.58	19.39	19.19	20.61	23.03	22.83	23.84
	Top-5 Accuracy	38.99	40.00	41.62	43.64	44.04	47.07	48.48	46.87
	Mean Top-5 Recall	32.06	33.68	32.91	37.21	36.54	39.28	41.66	37.15
Action	Top-1 Accuracy	10.91	11.11	11.72	11.72	13.54	14.95	14.55	15.15
	Top-5 Accuracy	26.26	27.07	29.90	31.72	33.54	32.12	34.95	34.75
	Mean Top-5 Recall	16.16	16.95	18.54	19.21	19.97	19.83	20.65	20.12
Mean TtA(5): VERB: 1.57 NOUN: 0.99 ACTION: 0.75									

Table 5.16: Validation results for SCP with 200 epochs, fusion

5.3.5 Validation Tables: Observation Time

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	31.72	30.91	32.73	32.93	34.34	34.95	36.97	35.96
	Top-5 Accuracy	72.12	71.31	72.53	73.13	71.52	71.92	72.12	73.54
	Mean Top-5 Recall	28.16	27.79	28.95	28.24	27.97	29.11	29.17	29.52
Noun	Top-1 Accuracy	15.96	15.15	16.16	16.57	19.80	17.98	20.81	22.02
	Top-5 Accuracy	36.36	36.57	35.96	38.18	39.39	39.80	42.42	44.24
	Mean Top-5 Recall	29.60	29.50	28.28	28.51	29.83	29.39	31.76	33.20
Action	Top-1 Accuracy	09.49	08.48	09.29	09.90	13.13	13.33	15.15	14.34
	Top-5 Accuracy	24.65	25.25	26.87	26.06	30.71	28.48	30.51	31.31
	Mean Top-5 Recall	14.53	15.11	15.60	14.31	17.11	16.94	17.70	17.82
Mean TtA(5): VERB: 1.53 NOUN: 0.93 ACTION: 0.73									

Table 5.17: Validation results for 2 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	30.91	29.49	32.73	30.51	33.94	35.56	35.96	33.33
	Top-5 Accuracy	70.51	71.11	72.93	71.72	71.72	71.31	71.11	70.91
	Mean Top-5 Recall	26.14	28.44	28.80	26.97	27.62	27.72	26.53	26.08
Noun	Top-1 Accuracy	15.35	13.74	15.96	17.17	17.58	18.18	21.21	20.81
	Top-5 Accuracy	34.14	36.77	36.36	40.40	39.80	39.39	42.02	44.24
	Mean Top-5 Recall	27.99	28.39	27.92	31.02	29.11	28.73	31.51	35.02
Action	Top-1 Accuracy	08.89	07.47	10.30	09.09	12.73	12.12	14.95	13.54
	Top-5 Accuracy	23.23	24.65	26.46	27.88	29.49	27.68	31.72	33.33
	Mean Top-5 Recall	13.92	14.81	15.86	15.32	17.37	15.71	17.44	18.22
Mean TtA(5): VERB: 1.53 NOUN: 0.94 ACTION: 0.72									

Table 5.18: Validation results for 4 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.49	27.47	31.92	31.72	33.13	34.95	36.97	34.95
	Top-5 Accuracy	69.09	70.51	71.92	71.11	71.11	71.92	72.12	73.13
	Mean Top-5 Recall	27.20	25.67	29.66	28.73	27.18	28.58	28.32	29.40
Noun	Top-1 Accuracy	16.57	16.57	17.58	17.98	19.60	17.98	21.41	20.20
	Top-5 Accuracy	35.96	37.78	35.96	38.59	40.00	40.81	42.02	43.43
	Mean Top-5 Recall	29.08	29.92	26.94	28.69	30.47	30.55	31.66	34.26
Action	Top-1 Accuracy	09.70	09.09	11.92	10.71	13.13	12.73	14.75	13.54
	Top-5 Accuracy	23.64	24.44	25.05	26.67	31.11	28.69	31.11	32.53
	Mean Top-5 Recall	13.34	14.92	14.16	14.26	17.45	15.94	17.02	17.52
Mean TtA(5): VERB: 1.51 NOUN: 0.92 ACTION: 0.70									

Table 5.19: Validation results for 6 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	31.92	31.52	33.33	33.33	33.54	35.76	36.36	34.95
	Top-5 Accuracy	70.71	71.52	72.93	72.93	71.92	71.11	70.51	71.11
	Mean Top-5 Recall	26.25	26.89	28.79	28.95	29.05	27.78	27.57	28.00
Noun	Top-1 Accuracy	16.77	16.77	16.36	16.57	19.39	18.99	20.20	22.02
	Top-5 Accuracy	35.56	36.36	34.55	37.17	38.99	38.18	41.21	42.42
	Mean Top-5 Recall	29.00	28.77	25.92	27.89	29.63	28.15	30.31	33.24
Action	Top-1 Accuracy	10.91	09.49	11.31	10.30	12.53	13.54	13.94	14.14
	Top-5 Accuracy	23.43	24.65	25.45	26.67	30.71	27.47	29.29	32.12
	Mean Top-5 Recall	13.55	14.14	14.58	15.04	17.06	15.20	15.58	17.17
Mean TtA(5): VERB: 1.53 NOUN: 0.90 ACTION: 0.69									

Table 5.20: Validation results for 8 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.48	30.91	31.72	30.30	32.73	34.95	37.37	35.35
	Top-5 Accuracy	70.51	69.09	71.72	70.30	68.08	70.51	71.11	70.51
	Mean Top-5 Recall	27.96	25.65	27.85	26.17	26.18	27.46	28.52	27.81
Noun	Top-1 Accuracy	15.15	16.16	16.77	18.79	18.38	18.59	21.21	22.42
	Top-5 Accuracy	35.56	38.18	36.16	40.61	39.80	39.80	41.01	42.83
	Mean Top-5 Recall	29.95	30.02	27.76	32.01	30.73	29.15	31.34	33.30
Action	Top-1 Accuracy	10.10	10.10	11.31	11.52	12.93	12.93	15.96	14.75
	Top-5 Accuracy	23.03	25.45	26.46	27.68	30.51	29.70	31.52	32.12
	Mean Top-5 Recall	13.34	14.29	14.36	14.85	16.31	16.10	16.98	17.35
Mean TtA(5): VERB: 1.51 NOUN: 0.94 ACTION: 0.71									

Table 5.21: Validation results for 10 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.89	29.29	32.12	29.09	32.53	32.53	33.54	32.93
	Top-5 Accuracy	69.90	69.70	70.51	71.72	70.91	70.71	71.92	71.52
	Mean Top-5 Recall	27.52	24.84	27.76	28.11	27.71	27.53	29.08	28.01
Noun	Top-1 Accuracy	16.16	16.36	16.36	16.57	18.99	18.38	20.20	20.00
	Top-5 Accuracy	35.35	38.18	38.18	40.00	38.99	41.62	41.41	42.42
	Mean Top-5 Recall	29.08	30.53	29.36	32.23	29.61	31.61	31.73	32.99
Action	Top-1 Accuracy	09.49	09.09	10.51	09.70	11.31	11.52	13.33	12.73
	Top-5 Accuracy	23.84	24.24	26.06	28.28	29.90	29.70	30.51	32.53
	Mean Top-5 Recall	13.08	13.20	14.60	15.44	16.49	16.33	16.05	17.60
Mean TtA(5): VERB: 1.52 NOUN: 0.91 ACTION: 0.69									

Table 5.22: Validation results for 12 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	31.11	28.28	31.11	31.11	33.54	34.95	35.35	33.94
	Top-5 Accuracy	71.52	71.11	72.32	72.53	70.91	70.51	72.12	71.11
	Mean Top-5 Recall	27.50	27.25	28.90	28.68	27.11	27.79	30.54	28.49
Noun	Top-1 Accuracy	17.17	16.97	17.58	17.78	19.19	18.18	20.61	21.01
	Top-5 Accuracy	35.56	37.78	36.36	41.21	40.61	40.00	42.22	42.02
	Mean Top-5 Recall	27.52	28.31	27.26	31.30	31.01	29.43	32.33	33.49
Action	Top-1 Accuracy	10.51	09.90	11.11	11.52	12.93	12.93	14.55	14.34
	Top-5 Accuracy	24.04	25.66	27.27	28.08	31.31	29.90	31.92	32.32
	Mean Top-5 Recall	13.33	15.47	15.69	15.47	18.58	16.46	17.59	18.09
Mean TtA(5): VERB: 1.53 NOUN: 0.91 ACTION: 0.72									

Table 5.23: Validation results for 16 encoding frames

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	30.10	29.29	32.93	30.51	32.53	34.75	35.76	34.14
	Top-5 Accuracy	70.10	70.71	71.31	70.91	69.29	68.28	68.48	70.91
	Mean Top-5 Recall	26.30	27.56	27.96	27.43	26.78	26.74	28.00	27.81
Noun	Top-1 Accuracy	16.16	15.15	16.97	16.77	17.78	16.77	18.79	20.00
	Top-5 Accuracy	34.55	35.56	34.95	37.58	36.77	38.79	39.19	39.80
	Mean Top-5 Recall	25.97	27.13	25.17	28.19	27.54	29.05	28.09	28.06
Action	Top-1 Accuracy	10.10	08.48	11.31	10.10	11.11	12.12	13.33	12.93
	Top-5 Accuracy	24.44	24.85	27.27	25.86	29.49	28.48	30.30	30.30
	Mean Top-5 Recall	13.63	13.93	14.60	14.15	16.56	16.09	16.06	16.26
Mean TtA(5): VERB: 1.51 NOUN: 0.89 ACTION: 0.70									

Table 5.24: Validation results for 24 encoding frames

5.3.6 Validation Tables: Other Modalities for 256 Cell Hidden Layer

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	26.46	28.69	25.66	30.91	29.49	31.92	33.33	31.72
	Top-5 Accuracy	73.13	71.92	72.73	73.74	73.13	73.74	74.34	73.74
	Mean Top-5 Recall	24.26	21.62	22.71	24.75	22.65	26.74	24.65	26.13
Noun	Top-1 Accuracy	09.70	08.48	09.29	08.48	07.88	11.11	11.92	10.91
	Top-5 Accuracy	24.85	25.25	24.44	27.07	24.04	25.86	26.67	30.51
	Mean Top-5 Recall	14.05	12.78	12.17	14.18	12.29	12.85	13.11	15.71
Action	Top-1 Accuracy	05.25	05.45	05.45	06.87	05.66	07.07	08.08	07.88
	Top-5 Accuracy	17.58	17.17	16.36	17.17	19.19	19.60	18.99	20.61
	Mean Top-5 Recall	06.96	06.56	05.55	06.05	07.05	07.80	07.07	08.41
Mean TtA(5): VERB: 1.52 NOUN: 0.68 ACTION: 0.51									

Table 5.25: Validation results for hidden layer with 256 cells, flow modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.09	30.51	30.30	28.89	29.49	30.71	30.71	30.51
	Top-5 Accuracy	74.14	72.73	72.93	73.33	72.12	73.33	71.31	71.52
	Mean Top-5 Recall	25.50	24.52	26.79	28.16	22.90	25.45	22.17	21.72
Noun	Top-1 Accuracy	16.36	16.16	17.78	19.19	19.80	20.00	21.62	19.39
	Top-5 Accuracy	38.79	38.38	39.19	39.60	40.81	42.83	42.63	42.02
	Mean Top-5 Recall	29.80	29.45	29.15	29.87	30.34	31.41	30.07	28.82
Action	Top-1 Accuracy	08.69	09.90	09.70	09.09	10.51	10.91	11.11	11.31
	Top-5 Accuracy	24.44	25.05	25.66	25.66	28.08	29.09	28.48	28.28
	Mean Top-5 Recall	13.75	14.26	14.38	13.40	14.95	14.68	14.25	13.55
Mean TtA(5): VERB: 1.52 NOUN: 0.92 ACTION: 0.64									

Table 5.26: Validation results for hidden layer with 256 cells, obj modality

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	31.31	31.92	33.54	31.72	33.13	36.57	37.58	35.35
	Top-5 Accuracy	72.93	72.32	73.94	73.94	74.55	72.93	73.74	72.32
	Mean Top-5 Recall	27.61	27.53	29.93	30.39	33.05	30.93	31.71	29.52
Noun	Top-1 Accuracy	17.58	16.57	18.79	20.20	21.01	20.81	22.83	22.02
	Top-5 Accuracy	41.41	41.01	40.61	43.84	44.85	45.05	46.87	45.45
	Mean Top-5 Recall	34.82	32.63	33.53	37.41	40.38	39.99	39.26	35.93
Action	Top-1 Accuracy	10.10	09.70	12.32	13.74	14.34	14.34	15.96	15.15
	Top-5 Accuracy	27.27	29.09	30.30	31.92	34.34	33.33	33.33	34.14
	Mean Top-5 Recall	15.14	17.61	17.78	19.07	20.62	19.20	18.64	19.00
Mean TtA(5): VERB: 1.55 NOUN: 0.99 ACTION: 0.76									

Table 5.27: Validation results for hidden layer with 256 cells, fusion

5.3.7 Validation Tables: Longer Observation Time 0.5s - 60s

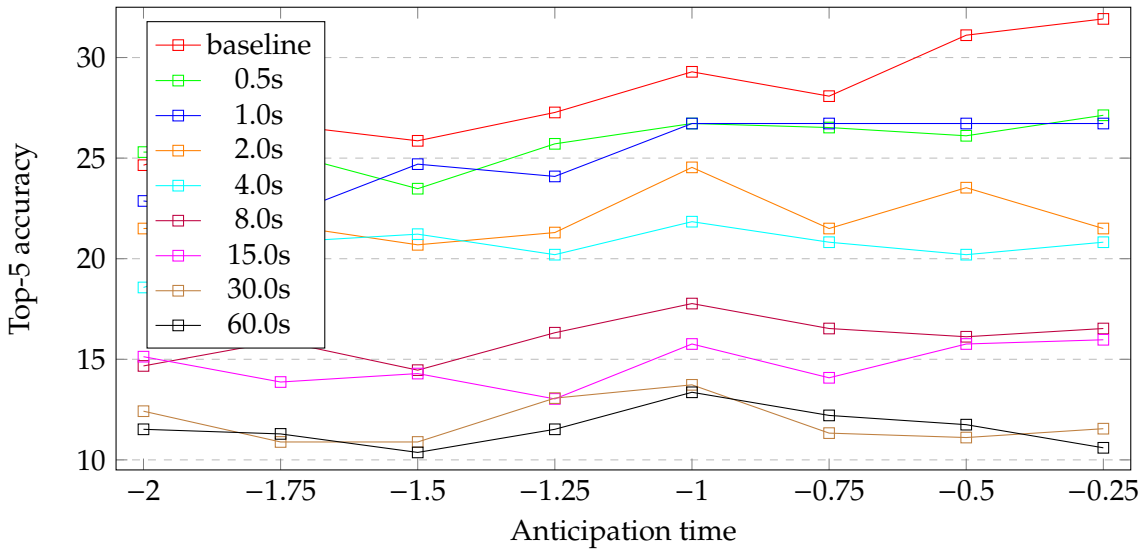

Figure 5.3: Comparison of time offsets for RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	30.97	32.39	30.97	29.15	30.57	29.35	33.20	35.02
	Top-5 Accuracy	71.26	72.47	70.85	71.86	71.46	72.06	71.46	71.46
	Mean Top-5 Recall	25.65	25.62	23.54	25.43	24.95	27.47	25.12	26.57
Noun	Top-1 Accuracy	16.60	16.19	14.98	15.59	15.38	15.59	17.61	17.00
	Top-5 Accuracy	37.65	35.43	33.81	36.64	37.04	39.47	39.07	39.27
	Mean Top-5 Recall	27.29	25.80	24.99	28.16	27.54	29.33	29.02	27.46
Action	Top-1 Accuracy	11.34	09.31	10.12	09.51	10.53	09.72	11.94	11.74
	Top-5 Accuracy	25.30	25.30	23.48	25.71	26.72	26.52	26.11	27.13
	Mean Top-5 Recall	13.72	14.65	13.35	14.30	14.28	14.65	14.21	14.67
Mean TtA(5): VERB: 1.52 NOUN: 0.91 ACTION: 0.69									

Table 5.28: Validation results for time offset 0.5s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.14	26.11	27.73	29.96	30.16	28.14	29.96	29.76
	Top-5 Accuracy	70.24	70.65	71.05	71.46	69.43	71.05	71.46	71.46
	Mean Top-5 Recall	25.41	26.67	25.91	25.90	24.35	25.35	25.45	26.26
Noun	Top-1 Accuracy	15.79	14.78	16.80	14.98	15.59	15.18	16.19	15.59
	Top-5 Accuracy	33.40	34.21	35.02	35.02	35.43	37.45	37.25	39.68
	Mean Top-5 Recall	23.41	24.46	24.88	23.62	25.52	27.59	26.72	29.21
Action	Top-1 Accuracy	09.72	09.92	10.53	09.31	09.31	09.31	09.72	09.72
	Top-5 Accuracy	22.87	22.06	24.70	24.09	26.72	26.72	26.72	26.72
	Mean Top-5 Recall	11.74	12.03	13.48	12.85	13.71	14.22	14.10	14.06
Mean TtA(5): VERB: 1.51 NOUN: 0.84 ACTION: 0.64									

Table 5.29: Validation results for time offset 1.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	24.54	27.38	24.54	27.79	28.19	28.19	29.41	29.82
	Top-5 Accuracy	69.37	70.18	69.98	71.20	70.18	71.20	71.20	72.01
	Mean Top-5 Recall	22.62	23.17	23.26	26.26	23.82	25.02	26.14	25.56
Noun	Top-1 Accuracy	13.79	14.00	14.81	15.01	14.60	14.40	15.21	15.01
	Top-5 Accuracy	32.45	32.25	33.87	32.86	34.08	33.87	34.28	33.47
	Mean Top-5 Recall	19.91	22.46	22.98	21.40	23.57	22.63	22.91	21.49
Action	Top-1 Accuracy	07.10	08.92	07.30	09.33	09.53	08.32	09.13	09.33
	Top-5 Accuracy	21.50	21.70	20.69	21.30	24.54	21.50	23.53	21.50
	Mean Top-5 Recall	10.31	11.23	09.89	10.30	12.50	10.62	11.82	10.53
Mean TtA(5): VERB: 1.48 NOUN: 0.81 ACTION: 0.57									

Table 5.30: Validation results for time offset 2.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	25.51	26.73	26.53	24.90	26.33	24.69	25.51	25.71
	Top-5 Accuracy	70.41	71.43	70.61	70.00	69.39	70.20	69.59	69.39
	Mean Top-5 Recall	22.54	24.69	22.30	22.08	21.22	21.71	22.73	21.14
Noun	Top-1 Accuracy	12.24	13.47	13.06	13.47	12.86	14.49	13.67	13.06
	Top-5 Accuracy	31.02	32.04	32.04	30.61	31.02	31.84	30.61	31.63
	Mean Top-5 Recall	19.65	21.00	19.64	19.29	18.37	19.22	17.78	17.95
Action	Top-1 Accuracy	07.14	07.35	07.96	07.14	07.55	08.16	08.57	07.55
	Top-5 Accuracy	18.57	20.82	21.22	20.20	21.84	20.82	20.20	20.82
	Mean Top-5 Recall	10.44	11.69	11.67	11.01	11.22	09.77	09.48	09.96
Mean TtA(5): VERB: 1.47 NOUN: 0.76 ACTION: 0.54									

Table 5.31: Validation results for time offset 4.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	22.31	22.52	20.87	21.90	21.90	20.66	23.55	21.69
	Top-5 Accuracy	67.77	68.80	69.42	69.21	69.63	69.83	69.63	69.63
	Mean Top-5 Recall	20.05	20.55	20.52	20.00	20.37	20.60	21.69	20.10
Noun	Top-1 Accuracy	10.12	09.30	08.06	09.71	09.92	10.54	10.95	10.74
	Top-5 Accuracy	25.41	26.24	27.27	27.89	27.48	28.93	28.31	26.65
	Mean Top-5 Recall	15.39	16.90	16.51	16.21	15.91	17.18	16.72	15.27
Action	Top-1 Accuracy	05.79	06.40	05.37	06.82	05.99	07.02	05.37	05.79
	Top-5 Accuracy	14.67	15.91	14.46	16.32	17.77	16.53	16.12	16.53
	Mean Top-5 Recall	07.56	08.32	07.02	07.63	08.66	07.60	08.31	07.67
Mean TtA(5): VERB: 1.44 NOUN: 0.67 ACTION: 0.42									

Table 5.32: Validation results for time offset 8.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	19.33	19.12	20.17	19.75	19.33	20.80	22.06	21.43
	Top-5 Accuracy	68.49	69.96	68.91	69.12	69.54	69.33	69.54	69.54
	Mean Top-5 Recall	19.95	21.24	20.24	20.23	21.07	21.17	20.76	20.85
Noun	Top-1 Accuracy	09.03	07.98	08.19	08.82	09.87	08.40	08.82	09.24
	Top-5 Accuracy	23.32	25.21	23.74	23.32	25.84	25.84	23.95	25.00
	Mean Top-5 Recall	13.74	14.70	14.89	14.07	15.08	15.85	14.09	14.49
Action	Top-1 Accuracy	03.57	02.73	04.41	04.20	03.78	03.78	03.99	04.41
	Top-5 Accuracy	15.13	13.87	14.29	13.03	15.76	14.08	15.76	15.97
	Mean Top-5 Recall	06.80	06.64	07.05	06.03	07.14	06.35	07.01	07.20
Mean TtA(5): VERB: 1.43 NOUN: 0.63 ACTION: 0.40									

Table 5.33: Validation results for time offset 15.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	20.48	21.57	20.04	20.70	22.22	20.92	20.04	19.83
	Top-5 Accuracy	68.41	67.97	66.45	66.67	67.32	66.88	66.88	66.88
	Mean Top-5 Recall	19.10	19.04	18.27	18.34	18.63	18.38	18.44	18.50
Noun	Top-1 Accuracy	05.88	06.75	06.97	06.54	07.41	06.75	06.75	07.41
	Top-5 Accuracy	22.00	22.44	23.75	23.53	22.88	22.44	21.57	23.97
	Mean Top-5 Recall	11.85	13.07	13.03	13.38	12.47	12.09	11.37	12.34
Action	Top-1 Accuracy	02.61	02.61	03.27	02.83	03.70	03.05	03.92	04.36
	Top-5 Accuracy	12.42	10.89	10.89	13.07	13.73	11.33	11.11	11.55
	Mean Top-5 Recall	04.58	04.29	04.35	05.87	05.53	04.20	03.88	04.10
Mean TtA(5): VERB: 1.38 NOUN: 0.59 ACTION: 0.34									

Table 5.34: Validation results for time offset 30.0s, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	15.90	17.51	17.74	17.74	17.05	18.20	17.51	19.82
	Top-5 Accuracy	68.20	68.43	67.97	68.20	68.20	67.97	67.28	68.66
	Mean Top-5 Recall	19.54	20.92	20.21	20.16	19.42	20.53	19.25	19.71
Noun	Top-1 Accuracy	05.53	04.15	05.53	03.92	04.15	05.07	05.30	04.84
	Top-5 Accuracy	20.28	18.66	21.20	20.28	20.51	20.97	20.74	19.12
	Mean Top-5 Recall	11.29	11.28	13.24	12.11	12.30	12.78	12.61	09.80
Action	Top-1 Accuracy	01.61	01.84	03.00	02.07	02.07	03.00	02.07	02.53
	Top-5 Accuracy	11.52	11.29	10.37	11.52	13.36	12.21	11.75	10.60
	Mean Top-5 Recall	05.61	06.04	05.19	06.33	06.94	06.95	05.77	04.53
Mean TtA(5): VERB: 1.40 NOUN: 0.57 ACTION: 0.35									

Table 5.35: Validation results for time offset 60.0s, RGB

Results for fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	27.33	30.16	30.57	30.77	31.58	32.39	32.39	33.00
	Top-5 Accuracy	71.26	72.47	73.08	72.27	72.47	72.27	72.67	71.66
	Mean Top-5 Recall	26.51	27.74	30.17	28.38	28.15	28.65	27.87	27.12
Noun	Top-1 Accuracy	18.62	18.22	16.19	18.02	17.00	18.22	19.23	19.23
	Top-5 Accuracy	39.07	40.49	39.88	42.91	43.72	43.52	43.52	42.11
	Mean Top-5 Recall	28.99	34.41	34.49	36.65	37.06	35.75	35.66	34.10
Action	Top-1 Accuracy	11.13	10.93	10.73	11.34	11.34	12.15	12.96	12.35
	Top-5 Accuracy	26.32	27.73	28.54	28.74	31.38	30.77	30.16	30.77
	Mean Top-5 Recall	15.69	16.60	17.34	17.02	18.32	18.32	18.24	17.40
Mean TtA(5): VERB: 1.53 NOUN: 0.96 ACTION: 0.70									

Table 5.36: Validation results for time offset 0.5s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	29.35	30.97	30.77	32.59	30.77	30.97	30.77	29.96
	Top-5 Accuracy	71.05	71.05	71.86	73.48	72.47	72.06	72.27	71.86
	Mean Top-5 Recall	24.58	25.15	26.77	27.43	26.45	26.51	25.82	25.03
Noun	Top-1 Accuracy	16.60	18.02	17.00	16.60	15.59	16.60	17.81	16.40
	Top-5 Accuracy	38.06	38.06	39.07	38.06	40.28	41.70	40.28	41.09
	Mean Top-5 Recall	27.81	28.13	28.15	28.24	30.13	31.93	30.63	30.40
Action	Top-1 Accuracy	09.51	10.32	10.12	10.12	09.51	09.92	11.54	10.32
	Top-5 Accuracy	24.90	24.70	25.51	26.52	28.14	28.74	29.35	29.55
	Mean Top-5 Recall	14.54	14.61	14.94	15.30	16.65	16.43	16.74	16.40
Mean TtA(5): VERB: 1.50 NOUN: 0.89 ACTION: 0.64									

Table 5.37: Validation results for time offset 1.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	24.75	27.38	24.75	27.59	28.80	28.40	29.01	28.80
	Top-5 Accuracy	68.97	70.59	71.40	70.99	71.81	71.60	72.21	71.40
	Mean Top-5 Recall	24.61	25.30	25.94	24.86	26.84	25.12	27.23	25.23
Noun	Top-1 Accuracy	15.62	15.21	17.44	17.04	16.63	17.44	18.05	16.84
	Top-5 Accuracy	35.29	35.70	37.12	36.71	39.35	38.34	39.55	38.74
	Mean Top-5 Recall	27.35	25.25	28.74	27.65	31.05	26.90	28.20	27.55
Action	Top-1 Accuracy	08.52	08.52	09.13	09.33	09.33	09.74	08.92	09.13
	Top-5 Accuracy	22.31	24.14	24.34	24.95	26.98	24.34	27.18	24.54
	Mean Top-5 Recall	14.00	14.26	13.69	14.73	15.76	14.08	16.11	12.64
Mean TtA(5): VERB: 1.49 NOUN: 0.87 ACTION: 0.60									

Table 5.38: Validation results for time offset 2.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	21.84	23.27	24.49	23.88	25.51	24.90	25.92	24.90
	Top-5 Accuracy	70.41	71.22	69.39	67.96	67.35	67.76	69.18	67.35
	Mean Top-5 Recall	22.20	26.70	23.19	22.88	22.59	21.62	23.05	20.99
Noun	Top-1 Accuracy	13.27	15.31	13.67	14.90	14.29	14.69	14.69	14.69
	Top-5 Accuracy	34.29	33.47	32.86	32.86	32.24	34.29	32.86	31.84
	Mean Top-5 Recall	22.45	23.16	22.00	22.70	22.06	22.95	22.24	20.66
Action	Top-1 Accuracy	06.53	07.96	06.73	08.37	06.53	07.35	07.96	07.14
	Top-5 Accuracy	22.04	21.43	23.06	21.22	23.67	22.04	22.24	21.63
	Mean Top-5 Recall	12.87	12.85	13.61	12.21	13.88	12.37	12.86	12.00
Mean TtA(5): VERB: 1.49 NOUN: 0.84 ACTION: 0.57									

Table 5.39: Validation results for time offset 4.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	18.39	19.21	18.60	18.60	21.49	19.21	23.35	22.31
	Top-5 Accuracy	66.94	65.50	66.94	66.32	67.56	68.60	67.15	68.18
	Mean Top-5 Recall	21.00	21.16	24.49	23.64	23.80	25.49	24.43	24.53
Noun	Top-1 Accuracy	08.47	09.30	09.71	09.71	11.16	09.30	09.92	09.71
	Top-5 Accuracy	27.27	27.48	26.65	28.31	27.89	30.99	27.07	26.65
	Mean Top-5 Recall	19.50	20.54	21.24	19.31	20.36	23.87	18.68	17.41
Action	Top-1 Accuracy	04.55	05.17	04.75	04.96	07.23	04.55	06.20	05.79
	Top-5 Accuracy	15.91	17.77	15.91	18.18	18.80	18.18	17.98	18.18
	Mean Top-5 Recall	09.53	11.68	10.65	11.34	11.57	11.67	10.71	10.97
Mean TtA(5): VERB: 1.43 NOUN: 0.71 ACTION: 0.46									

Table 5.40: Validation results for time offset 8.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	19.96	18.28	19.54	18.49	19.33	20.38	21.43	21.01
	Top-5 Accuracy	67.44	66.60	67.23	67.65	69.12	68.70	68.28	68.49
	Mean Top-5 Recall	21.86	20.39	20.40	20.68	20.58	21.03	20.73	20.70
Noun	Top-1 Accuracy	08.40	08.19	09.45	08.61	09.03	10.29	09.03	09.03
	Top-5 Accuracy	25.63	25.00	25.63	23.32	26.26	24.37	24.37	25.21
	Mean Top-5 Recall	15.88	16.30	17.34	15.87	18.50	17.03	15.95	17.16
Action	Top-1 Accuracy	04.83	03.36	05.25	04.41	04.83	05.25	05.04	04.62
	Top-5 Accuracy	13.87	12.18	14.29	14.08	15.76	14.92	13.66	14.71
	Mean Top-5 Recall	07.83	06.44	07.78	07.93	09.24	08.34	07.24	07.51
Mean TtA(5): VERB: 1.43 NOUN: 0.67 ACTION: 0.39									

Table 5.41: Validation results for time offset 15.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	20.26	21.35	20.48	21.13	21.79	21.13	20.26	19.39
	Top-5 Accuracy	68.63	67.76	66.88	66.67	67.54	66.88	66.67	67.76
	Mean Top-5 Recall	19.16	18.94	18.40	18.30	18.70	18.38	18.34	18.81
Noun	Top-1 Accuracy	06.10	07.19	06.75	06.75	06.97	07.19	06.75	07.19
	Top-5 Accuracy	22.44	23.09	23.97	23.97	23.09	22.66	22.22	23.53
	Mean Top-5 Recall	12.08	13.40	13.10	13.52	12.73	12.13	11.94	12.13
Action	Top-1 Accuracy	02.40	03.05	02.83	02.83	03.49	03.49	03.92	04.14
	Top-5 Accuracy	12.64	11.76	11.55	13.29	13.94	11.55	11.11	11.11
	Mean Top-5 Recall	04.65	04.79	04.76	05.99	05.68	04.28	04.30	04.00
Mean TtA(5): VERB: 1.39 NOUN: 0.60 ACTION: 0.35									

Table 5.42: Validation results for time offset 30.0s, fusion

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	16.82	17.97	16.82	17.28	17.97	16.59	17.28	18.89
	Top-5 Accuracy	67.51	67.28	67.97	67.74	67.51	67.05	67.28	68.20
	Mean Top-5 Recall	20.65	20.91	22.09	20.12	20.66	20.26	19.43	19.67
Noun	Top-1 Accuracy	04.84	05.30	05.30	04.61	04.61	04.61	04.84	04.61
	Top-5 Accuracy	19.82	18.43	19.59	19.82	19.82	20.28	20.05	19.12
	Mean Top-5 Recall	10.71	11.31	12.29	12.01	11.95	12.67	12.47	09.94
Action	Top-1 Accuracy	01.61	01.84	02.76	02.07	02.07	01.84	02.30	02.07
	Top-5 Accuracy	09.68	10.60	11.29	12.21	13.36	11.75	11.75	09.91
	Mean Top-5 Recall	04.64	05.63	06.12	06.44	07.09	06.58	05.71	04.22
Mean TtA(5): VERB: 1.39 NOUN: 0.56 ACTION: 0.34									

Table 5.43: Validation results for time offset 60.0s, fusion

5.3.8 Validation Tables: Full EK-55 Dataset Results

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	26.13	26.77	28.22	29.08	31.07	32.56	33.89	34.63
	Top-5 Accuracy	75.00	75.54	76.15	77.27	77.76	77.98	78.84	79.36
	Mean Top-5 Recall	36.95	38.00	38.32	40.22	41.75	42.40	43.30	43.99
Noun	Top-1 Accuracy	17.08	17.56	17.96	18.83	20.51	21.58	22.37	23.09
	Top-5 Accuracy	41.11	41.93	43.44	45.47	46.64	47.71	49.58	50.48
	Mean Top-5 Recall	39.37	40.02	41.44	43.34	45.11	45.54	47.73	48.45
Action	Top-1 Accuracy	10.00	10.44	11.18	11.75	13.01	14.26	14.82	15.55
	Top-5 Accuracy	25.20	26.43	27.88	29.36	31.07	31.96	33.95	35.08
	Mean Top-5 Recall	09.56	10.40	11.15	11.72	12.44	12.73	13.58	13.83
Mean TtA(5): VERB: 1.60 NOUN: 1.02 ACTION: 0.69									

Table 5.44: Validation results for full dataset with 1024 hidden cells, RGB

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	26.65	27.53	28.40	29.02	30.27	32.10	33.39	34.79
	Top-5 Accuracy	74.07	74.66	75.06	75.72	75.95	76.55	76.69	77.29
	Mean Top-5 Recall	31.13	31.30	33.15	33.62	33.96	36.03	34.90	36.35
Noun	Top-1 Accuracy	10.86	11.58	11.46	12.41	13.01	13.72	14.62	15.59
	Top-5 Accuracy	29.47	31.15	31.66	32.48	34.03	35.70	36.71	36.95
	Mean Top-5 Recall	23.62	26.35	26.98	27.36	28.12	29.69	30.01	30.12
Action	Top-1 Accuracy	06.54	06.78	06.98	07.86	08.11	08.93	09.69	10.02
	Top-5 Accuracy	17.06	18.02	18.64	19.99	21.52	22.18	23.31	24.36
	Mean Top-5 Recall	04.07	04.74	05.11	05.21	06.08	05.87	06.29	06.39
Mean TtA(5): VERB: 1.57 NOUN: 0.80 ACTION: 0.52									

Table 5.45: Validation results for full dataset with 1024 hidden cells, flow

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	25.99	27.33	27.94	27.80	28.50	29.08	29.81	29.79
	Top-5 Accuracy	75.86	76.11	76.97	77.13	77.37	77.96	78.28	78.26
	Mean Top-5 Recall	34.90	34.33	36.49	36.59	37.20	37.94	37.98	37.29
Noun	Top-1 Accuracy	18.95	19.33	20.27	20.98	21.44	21.96	23.27	24.30
	Top-5 Accuracy	43.72	45.43	46.62	47.99	50.00	50.78	51.81	52.84
	Mean Top-5 Recall	41.61	43.21	44.91	46.15	48.60	49.12	49.71	50.16
Action	Top-1 Accuracy	08.67	09.35	09.57	09.69	10.40	10.80	11.87	12.33
	Top-5 Accuracy	24.76	25.93	27.41	28.70	30.11	30.67	31.88	32.60
	Mean Top-5 Recall	09.37	09.97	10.56	10.86	11.09	11.30	11.68	11.87
Mean TtA(5): VERB: 1.59 NOUN: 1.08 ACTION: 0.67									

Table 5.46: Validation results for full dataset with 1024 hidden cells, obj

Class	Metric	Anticipation time							
		2.00s	**1.75s**	**1.50s**	**1.25s**	**1.00s**	**0.75s**	**0.50s**	**0.25s**
Verb	Top-1 Accuracy	28.46	29.59	30.25	31.80	32.52	34.69	36.50	37.25
	Top-5 Accuracy	76.97	77.78	78.40	78.78	79.38	79.97	80.21	80.81
	Mean Top-5 Recall	39.19	40.30	41.60	42.17	42.99	44.03	45.34	46.39
Noun	Top-1 Accuracy	19.75	20.41	20.94	22.24	23.25	24.44	25.34	26.39
	Top-5 Accuracy	46.68	47.89	49.40	50.64	51.61	52.74	54.51	55.03
	Mean Top-5 Recall	45.82	46.49	48.29	49.42	50.08	50.62	52.15	52.50
Action	Top-1 Accuracy	11.18	11.83	12.51	14.06	14.80	16.11	16.67	17.66
	Top-5 Accuracy	29.34	29.87	31.98	33.31	34.73	35.66	37.51	38.54
	Mean Top-5 Recall	11.71	12.59	13.37	14.17	14.70	15.13	15.78	16.06
Mean TtA(5): VERB: 1.63 NOUN: 1.11 ACTION: 0.76									

Table 5.47: Validation results for full dataset with 1024 hidden cells, fusion

References

[1] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *ICLR*. 2021.

[2] Dima amen et al. *EPIC-KITCHENS-100- 2022 Challenges Report*. Tech. rep. University of Bristol, 2022.

[3] Oliver Cieplinski. *Exploring egocentric action anticipation using RU-LSTMs*. 2022.

[4] Dima Damen et al. "Scaling egocentric vision: The epic-kitchens dataset". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 720–736.

[5] Dima Damen et al. "The epic-kitchens dataset: Collection, challenges and baselines". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2020), pp. 4125–4141.

[6] Antonino Furnari. *Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video*. `https://github.com/fpv-iplab/rulstm`. [Online; Accessed 18-March-2023]. 2020.

[7] Antonino Furnari and Giovanni Maria Farinella. "Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2020).

[8] Antonino Furnari and Giovanni Maria Farinella. "What Would You Expect? Anticipating Egocentric Actions with Rolling-Unrolling LSTMs and Modality Attention." In: *International Conference on Computer Vision*. 2019.

[9] Rohit Girdhar and Kristen Grauman. "Anticipative Video Transformer". In: *ICCV*. 2021.

[10] Xiao Gu et al. *TransAction: ICL-SJTU Submission to EPIC-Kitchens Action Anticipation Challenge 2021*. 2021. arXiv: `2107.13259 [cs.CV]`.

[11] Zeyu Jiang and Changxing Ding. *1st Place Solution to the EPIC-Kitchens Action Anticipation Challenge 2022*. 2022. arXiv: `2207.05730 [cs.CV]`.

[12] Yin Li, Miao Liu, and Jame Rehg. "In the eye of the beholder: Gaze and actions in first person video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[13] Will Price and Dima Damen. "An evaluation of action recognition models on epic-kitchens". In: *arXiv preprint arXiv:1908.00867* (2019).

[14] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[15] Tsung-Ming Tai et al. *Higher Order Recurrent Space-Time Transformer for Video Action Prediction*. 2021. arXiv: 2104.08665 [cs.CV].

[16] Tsung-Ming Tai et al. *Unified Recurrence Modeling for Video Action Anticipation*. 2022. arXiv: 2206.01009 [cs.CV].

[17] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[18] Zeyun Zhong et al. *Anticipative Feature Fusion Transformer for Multi-Modal Action Anticipation*. 2022. arXiv: 2210.12649 [cs.CV].

Acknowledgments