Leveraging Next-Active Objects for Context-Aware Anticipation in Egocentric Videos

Sanket Thakur[1,4], Cigdem Beyan[2], Pietro Morerio[1], Vittorio Murino[3, 1], and Alessio Del Bue[1]

[1]Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT)
[2]Department of Management, Information and Production Engineering, University of Bergamo, Dalmine, Italy
[3]Department of Computer Science, University of Verona, Italy
[4]Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture (DITEN), University of Genoa, Italy

Abstract

Objects are crucial for understanding human-object interactions. By identifying the relevant objects, one can also predict potential future interactions or actions that may occur with these objects. In this paper, we study the problem of Short-Term Object interaction anticipation (STA) and propose NAOGAT (Next-Active-Object Guided Anticipation Transformer), a multi-modal end-to-end transformer network, that attends to objects in observed frames in order to anticipate the next-active-object (NAO) and, eventually, to guide the model to predict context-aware future actions. The task is challenging since it requires anticipating future action along with the object with which the action occurs and the time after which the interaction will begin, a.k.a. the time to contact (TTC). Compared to existing video modeling architectures for action anticipation, NAOGAT captures the relationship between objects and the global scene context in order to predict detections for the next active object and anticipate relevant future actions given these detections, leveraging the objects' dynamics to improve accuracy. One of the key strengths of our approach, in fact, is its ability to exploit the motion dynamics of objects within a given clip , which is often ignored by other models, and separately decoding the object-centric and motion-centric information. Through our experiments, we show that our model outperforms existing methods on two separate datasets, Ego4D and EpicKitchens-100 ("Unseen Set"), as measured by several additional metrics, such as time to contact, and next-active-object localization. The code can be found on project page : sanketsans.github.io/wacv24

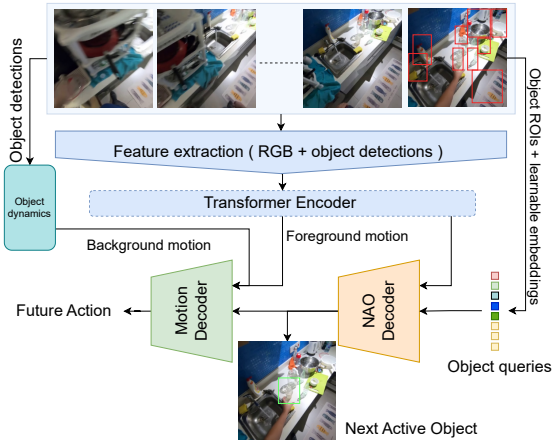

Figure 1. Our proposed model, NAOGAT, uses both features from video frames and object detections, which are combined and fed to a transformer encoder. Given object queries and encoded frame features from last observed frame, NAO decoder predicts relevant next-active-object detections. This information from NAO decoder is then passed along to the Motion decoder, which utilizes object dynamics to extract background information as object trajectories, to anticipate future frame interaction with encoder frame features (foreground motion) and next-active-object decoded features.

1. Introduction

Have you ever wondered how humans are able to effortlessly navigate their surroundings and perform actions based on what they see, especially in virtual reality (VR) and augmented reality (AR) environments ? Such actions often involve contact with objects which are referred to as *active objects* in the egocentric vision literature [32]. Understanding and predicting the interactions with these objects are essential for enhancing the realism and interactiv-

ity of VR and AR experiences [17, 29].

For example, Fig. 1 shows a first-person video clip where someone is about to perform a specific action. From the observed frames, it is reasonable to guess that the person is going to make a contact with the glass to possibly perform a *wash* or *fill* action. This reasoning helps us to recognize two important cues from a video: (1) which object will be "used" (i.e., active) in the future, and (2) what possible actions can be performed with that object. The category of objects significantly influences the nature of actions performed on them [1]. For example, a *cut* action might not be performed in this scenario if we know that *glass* is the next-active-object. Thus, intuitively, a model can make better predictions if it is able to anticipate which object(s) in the scene is possibly engaged in the very next future, to support and drive the identification of the future action. This can enable more immersive and interactive experiences, where users can seamlessly interact with virtual objects based on anticipated actions, leading to enhanced user engagement and satisfaction.

This concept is particularly relevant in the Short-Term Anticipation (STA) task, which involves predicting the next-active-object (NAO) and its position, along with the time to contact (TTC) with that object, as well as the upcoming action, for a given video clip. This task depends on the assumption that the NAO is visible or present in the last observed frame, enabling its identification and localization [15]. The task that is more frequently exploited in the egocentric vision literature is instead action anticipation [13, 39], which refers to predicting a future action involving an object interaction without necessarily requiring the object to be visible in the last observed frame.

The use of object-centric cues has shown significant promise in various video understanding tasks, such as action recognition [18, 31, 38], hand-object forecasting [8, 22, 24, 30], and action anticipation [12, 23, 34]. However, egocentric action anticipation methods [12, 13, 23, 39] often overlooked such cues and mostly relied on holistic scene features and/or hand features. Indeed for STA, there exists no method explicitly considering the information that could be gained from the object and more importantly NAOs.

In this paper, we propose a novel multi-modal architecture, called NAOGAT (Next-Active-Object Guided Attention Transformer), that involves training a model to attend to the objects in the last observed frame based on an obsered video clip, allowing it to predict the next-active-object (NAO). By incorporating the most relevant objects in the upcoming action and modeling the object dynamics of an observed clip, our model can make more accurate predictions of future actions and time to contact with those object(s) (NAO) and improve its overall performance.

The experimental analysis performed on two large-scale datasets: Ego4D [15] and EpicKitchen-100 (EK-100) [4] demonstrate the favorable performance of NAOGAT with respect to several other methods, and indeed prove the importance of NAO cues and object dynamics for the targeted task. Particularly, on the Ego4D dataset [15], we show a 2.16% gain in Average Precision for VERB + NAO, while a notable improvement of 7.33% is observed in estimating the TTC + NAO.

The contributions of this paper can be summarized as follows:

- We present a Transformer-based method, called NAOGAT, for STA task, which models next-active-object anticipation as a fixed-set prediction problem based on object queries.
- We propose a joint learning strategy based on fixed and learnable object queries which are extracted as ROIs from object detections and learned based on global context of video respectively: relevant detections for next-active-object guide the model to anticipate for object-specific future actions.
- The proposed method also exploit the motion dynamics of objects in sampled frames to model background information, in terms of object trajectories, along with foreground motion extracted from RGB features to better represent the human-object interaction.
- We provide next-active-object annotations for the EpicKitchen-100 [4] dataset in terms of NAO location *w.r.t* last observed frame, which can be used to further advance research in egocentric video analysis and action anticipation.

2. Related Work

Existing action anticipation methods in egocentric videos have primarily focused on utilizing features extracted from video clips, but they often overlooked the importance of objects and their interactions. Below, we discuss the existing works on the next-active-object and overall action anticipation methods in first-person videos since in this paper we aim to show that leveraging the next-active-objects would improve the action anticipation task.

Next-active-object. In egocentric vision literature, *active* and *passive* objects were first time introduced by Pirsiavash and Ramanan [32], which defined the active objects as those the first-person is interacting with, while passive objects stand for the contrary. Dessalene et al. [7] used the [32] description and proposed a method to predict *next-active-objects*, however, they limited their objective only to the objects that are contacted by the first-person's *hand*. Consequently, this approach requires the hands and the next-active-objects to be visible in the frames, which might be a significant limitation in practical applications. Furnari et al. [11] utilized object tracking to detect the next-active-objects, limited to being able to predict them only in *one*

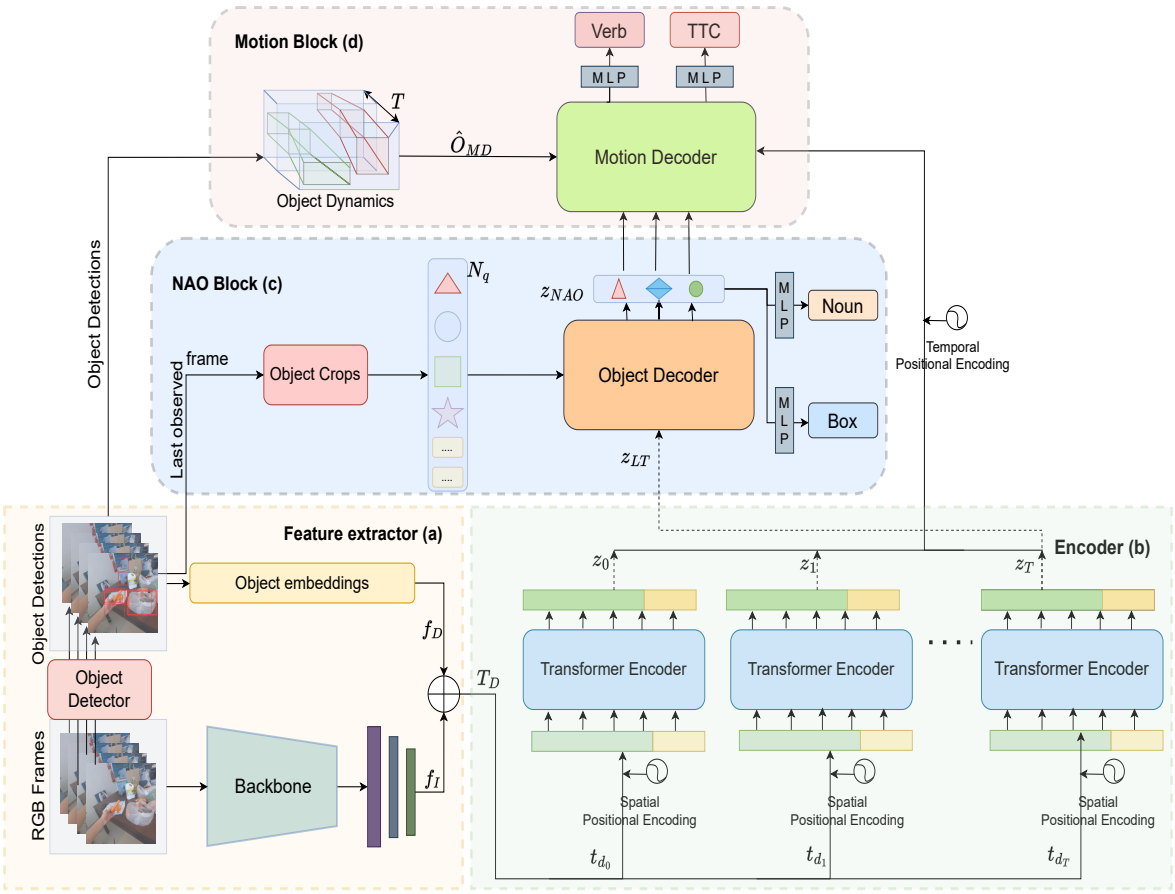

Figure 2. Our NAOGAT model first extracts feature information and object detections from a set of frames within an observed clip segment by means of a backbone network and an object detector; object detections are then transformed into object embeddings using an MLP network. The frame features are then concatenated with object embeddings to be sent to the transformer encoder after appending with spatial positional encoding. The encoder then extracts foreground motion (video memory) and global context features, which are used in two separate decoders to perform object-centric and motion-centric predictions. For object decoder, detections from the last observed frame and learnable embeddings are used to create object queries to perform fixed-set predictions for the NAO class label and its bounding box using a transformer decoder. In the last stage, we leverage Object dynamics to extract background motion in terms of object trajectories for detected objetcs in sampled frames. We then use the object decoder's outputs with the combined frame representation of video memory and object dynamics to perform predictions for motion-related outputs, such as future action and time to contact (TTC).

future frame. Other works [20, 24] proposed to identify the objects either in a single image (without analyzing the spatiotemporal data) or by predicting the future hand motion. Both fall behind in explicitly exploiting the future active object information to predict future action. Recently, Thakur et al. [37] proposed a transformer-based approach that applies collaborative modeling of RGB and object features to anticipate the location of the next-active-object(s) several frames ahead of the last observed frame.

Action Anticipation. This task stands for forecasting the future actions of a person given an egocentric video clip including the past and current frames. Several approaches in this line have focused on learning the scene features with Convolutional Neural Networks (CNNs), e.g., by modeling the hand-object contact points [23]. Others aggregated the past contextual features [10, 12, 36], and a few focused on modeling the future interaction of consecutive frames [40]. With the emergence of Vision Transformers [9, 28], researchers have started to investigate the utility of transformers in their work. For instance, [13] proposed causal modeling of video features, introducing sequence modeling of frame features to decode the interactions in consecutive future frames. We et al. [39] proposed a long-term understanding of videos using Multiscale Transformers by hierarchically attending to previously cached memories. In a recent work, Zhang et al. [41] proposed fusing object information along with RGB frame features for a better video context representation in addition to investigating the impact of the audio modality for action anticipation. Our work distinguishes itself from the related work by addressing action anticipation in egocentric videos with the prediction of the next-active-objects.

3. Method

As already mentioned, the Short-Term Anticipation (STA) task aims at predicting the next human-object interaction happening after a certain (unknown) time δ, named the Time To Contact (TTC), relying on evidence up to time T. Thus our model's input is a T-frames video sequence $V = \{v_i\}_{i=1}^{T}$, where $v_i \in \mathbb{R}^{C \times H_o \times W_o}$ is an RGB frame, while it must output 4 unknowns at time T, following the protocol introduced in [15]: the NAO noun class ($\hat{n}$), its location, i.e. bounding box ($\hat{b}$), a verb depicting the future action ($\hat{v}$) and the time to contact ($\hat{\delta}$), which estimates how many seconds in the future the interaction with the object will begin.

3.1. Method Overview

We now introduce our model architecture, as illustrated in Fig. 2, which is designed to predict the class and location of the next-active-object and the time required to make contact with the object (the TTC δ) as well as the future action, based on a given observed clip. The proposed method comprises 4 main modules. First, a *feature extractor* (Fig. 2 (a) , Sec. 3.2) operates on frames from a sampled video clip to extract RGB and object detection features. This is followed by an *Encoder Block* (Fig. 2 (b) , Sec. 3.3) that operates on the combined features to facilitate the exchange of information across frames. Following the encoder, the two separate head architectures - *NAO Block* (Fig. 2 (c) , Sec. 3.4) and *Motion Block* (Fig. 2 (d) , Sec. 3.5), is used to predict the next-active-object information and future action prediction respectively. Our model employs object queries from object detection in the last observed frame to locate and identify the next-active-object and is inspired by the direct set prediction problem [2], which involves predicting a fixed set of objects and modeling their relationship. In addition, we leverage object dynamics to extract background motion in a video clip. This involves incorporating object trajectories for detected objects in sampled frames within the *Motion Block* and utilizing attended frame features from the encoder module to model the relationship with NAO priors (Fig. 2-(c)), in order to predict future action and TTC. An overview of our model architecture in shown in Fig. 1, which is described in detail in Fig. 2. In the following we describe each model component in detail, followed by training and implementation details.

3.2. Feature extractor

For a given video clip, we sample a set of T frames $V = \{v_i\}_{i=1}^{T}$, where, $v_i \in \mathbb{R}^{C \times H_o \times W_o}$, which are fed to *i)* a backbone network for feature extraction *ii)* a pre-trained object detector (Fig. 2-(a)).

i) While a number of video-based backbone architectures have been proposed [3, 9, 18, 28, 39] to extract frame-level feature representation from a given video clip, for the task of STA a suitable spatial-temporal encoder is required to be able to extract both static appearance information (*e.g.*,, object location, size) and motion cues. Video Swin Transformer [28], a recently proposed spatial-temporal transformer architecture was proposed based on shifted windows architecture of Swin Transformer [25] for the video domain. Video Swin contains just a single temporal downsampling layer and can be easily adapted to output per-frame feature maps, essential for us to localize and identify the next-active-objects at the same time reasoning on the whole sequence. Specifically, we adopted the Swin-T [28] architecture as our backbone. The frames V are given in parallel to the video swin architecture to extract frame features, $f_I \in \mathbb{R}^{T \times H \times W \times C^{'}}$, where T, H and W denotes the temporal length of video clip, as well as height and width of the feature maps respectively.

ii) In addition, a Faster R-CNN [35] based object detector pre-trained on Ego4D [15], is used to extract the object detections from the sampled frames, in terms of bounding boxes coordinates and confidence score, resulting in a $(4+1)$-dimensional vector. We limit the number of bounding boxes to be used to a fixed number Q to maintain a consistent number of detections across each frame. If there are fewer detections than Q, then dummy coordinate and score values corresponding to no detection are appended. Finally an MLP processes 5-d vectors into object embeddings $f_D \in \mathbb{R}^{T \times Q \times D}$.

Finally, visual features are also projected to a shared dimension D as of f_D, using a 2D convolution layer with kernel size as 1, $f_I \in \mathbb{R}^{T \times H \times W \times D}$. Finally, the features from each modality are then flattened and *separately* concatenated along the temporal dimension, producing a set of $T_D = \{t_{di}\}_{i=1}^{T}$ multimodal embeddings, where $t_{di} \in \mathbb{R}^{(H \times W + Q) \times D}$.

3.3. Transformer Encoder

In the next step, according to Fig. 2-(b), the concatenated multimodal embeddings, T_D are simultaneously passed to a Transformer Encoder [2] after appending spatial positional encoding. The Transformer Encoder blocks allow exchanging frame-level information within inter-frame features while maintaining the same dimension. The output of the encoder is Z, where $Z \in \mathbb{R}^{T \times (H \times W + Q) \times D}$, which is the combined features representation across frames and object detections. It is split into 2 parts : 1) Global-context memory, z_{LT}, where $z_{LT} \in \mathbb{R}^{(H \times W + Q) \times D}$ extracted from last frame of Z, 2) Video-only memory: $\{z_i\}_{i=1}^{T}$ where $z_i \in \mathbb{R}^{H \times W \times D}$, which aims to captures foreground motion cues such as *e.g., hands in first-person vision* [13]. The Global-context memory and Video-only memory are then used by the NAO and Motion blocks to find the instances that corresponds to possible next-active-object and also anticipate the future action respectively.

3.4. NAO Block

The STC task requires anticipating the location of the next-active-object wrt the *last* frame observed by the model. Therefore, as shown in Fig. 2-(c), we only use the features corresponding to the last frame from the transformer encoder, namely z_T, as input to the NAO block, along with N_q object queries for the frame of interest. We define our object queries as the regions of interest (ROIs) extracted by the object detector, i.e., a feature map for each detection. If there are no sufficient detections, *i.e.,* the number of detections is less than N_q, then we append learnable tokens for the rest of the queries. Our object decoder follows the standard architecture of the transformer decoder [2], transforming N_q embeddings of size D using multi-headed attention mechanisms. The N_q object queries are decoded by using z_T as key/value pairs in the multiple multi-head attention layers. The decoded features, $z_{NAO} \in \mathbb{R}^{N_q \times D}$, are then used to predict bounding box coordinates ($\hat{b}$) and class labels ($\hat{n}$) by an additional MLP block, resulting in N_q final predictions for the next-active-object. The decoder's primary function is to attend to objects (detected/learned) in the last observed frame based on a global context of a video clip, resulting in the prediction of a possible next-act-object and its corresponding object label.

3.5. Motion Block

Object Dynamics. We propose to integrate the video frame features from the transformer encoder with the object dynamics of detected objects in the video clip, in order to better estimate the time required to approach the next-active-object predicted by our object decoder (Sec. 3.4). As shown in Fig. 2-(d), object dynamics refer to a proxy for object trajectories of background objects in the video clip. [18] previously used object dynamics to enhance the frame representation for effective motion information modeling in videos. However, their approach requires object region information and multiple stacking of the feature representation block in a transformer encoder for action recognition benchmark. In contrast, we treat object motion dynamics as a separate module for extracting object traversals. Object trajectories are the bounding box movement across the frames in sampled video clip. We interpret these trajectories as background motion because they correspond to passive motion in the scene, and combine them with transformer encoder outputs to model human-object motion features. This approach has proven useful in estimating the speed of interaction and predicting motion-centric information. The Object Dynamics block takes as input the object detection's box locations od_i for a frame i and outputs spatial-temporal tokens,

$$o_{\hat{MD}0}, \dots o_{\hat{MD}T} = \textit{OMD}(od_0, \dots, od_T) \qquad (1)$$

where $o_{\hat{MD}i} \in \mathbb{R}^{H \times W \times D}$. In OMD, initially, each object detection is expanded from $T \times Q \times 4$ into $T \times Q \times D$ tokens using a MLP. These tokens are flattened then used to perform self-attention operation and projected on a spatial-temporal dimension $THW \times D$ using a bi-linear interpolation sampler operation [19] to output object trajectories for frames used in the inputs $o_{\hat{MD}i}$. This module provides more detailed information on the object motion, *i.e.;* background motion of frames.

Motion Decoder. It was empirically observed that a single Object Decoder (Sec. 3.4) leads to the dropping of motion information across the frames, resulting in very poor performance for future action prediction. For this purpose, we decided to use a separate decoder for motion-related predictions (verb and TTC). Inspired from [13], we additionally combine frame features with the object motion dynamics features to model the foreground motion from video memory (Sec. 3.3) and background motion from object dynamics (Sec. 3.5) at the frame level.

$$z'_0, \dots z'_T = MLP(LN(z_0, \dots, z_T \bigoplus o_{\hat{MD}0}, \dots, o_{\hat{MD}T})) \qquad (2)$$

Here, object motion dynamics features, $o_{\hat{MD}i}$ are added to encoder features z_i along spatial and temporal dimension *T, H, W*, where $\bigoplus$ denotes such element-wise summation. This is followed by a Layer Norm (LN) and a MLP. In addition, to influence our future action prediction based on our next-active-object prediction, we add the object decoder embeddings, z_{NAO} to the last observed frame before feeding the sequence to the decoder.

$$\hat{z_1}, \hat{z_2}, \dots, \hat{z_{T+1}} = \textit{D}(z'_0, z'_1, \dots, z'_T + z_{NAO}) \qquad (3)$$

We implement D using the masked transformer decoder as followed in popular approaches such as [33]. We feed the modified inputs features to the masked decoder after appending with temporal positional encoding. The masking ensures that the model attends to specific parts of the input while performing the prediction for the next consecutive position. That helps our model to understand the interaction of person and the surrounding motion. The additional input of z_{NAO} helps to refine future action prediction. The design differs considerably from [13], since we model the background and foreground motion in a combined fashion with additional priors of next-active-object added to last observed frame features before the causal modeling. The decoder network D is designed to produce attentive features corresponding to the future frames using the object motion dynamics and also the next-active-object information in the last observed frame to anticipate the future action. We use the future frame feature, z_{T+1} to predict future action label $\hat{v}$ and the TTC δ corresponding with the next-active-object obtained from the object detector, using a feed-forward network.

3.6. Training

Let us denote y as the set of ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ as the set of N predictions, relating to N

object queries. Based on the procedure of finding matching elements of [2], we identify the one-to-one matching for the predictions with the ground truth labels using the *Hungarian loss* for all pairs matched.

Bounding box loss. The major difference between us and [2] is that we aim to learn bounding boxes based on some initial guesses, rather than only performing the predictions directly. The predicted bounding boxes are regressed using a combination of L1 loss and the generalized IoU loss and are defined as :

$$\mathcal{L}_{box} = \lambda_{iou}\mathcal{L}_{iou}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{L1}||b_i - \hat{b}_{\sigma(i)}||_1 \quad (4)$$

where $\lambda_{iou}, \lambda_{L1} \in \mathbb{R}$ are hyper-parameters.

Classification losses. The second loss, denoted by $\mathcal{L}_{noun}$ and $\mathcal{L}_{verb}$, is a cross-entropy loss that supervises the prediction of labels for the next-active-object and the future action:

$$\mathcal{L}_{verb/noun}(\hat{y_i}, y_i) = \sum_{t=0}^{N} y_i^t . \log(\hat{y}_i^t) \quad (5)$$

Regression and feature Loss. The regression loss, denoted by $\mathcal{L}_{ttc}$ is the smooth L1 loss [14] and is used to train the model to regress the time to contact prediction. Finally, we also use a feature loss, L_{feat} defined below in Eq. 6 from [13] which aims at leveraging the predictive structure of the motion decoder 3.5: the decoder is basically trained to predict future frame features given frames up to time t only.

$$\mathcal{L}_{feat} = \sum_{t=0}^{N} ||\hat{z}_{t+1} - z'_{t+1}||_2^2, \quad (6)$$

In the end, all losses are combined to produce the overall loss:

$$\mathcal{L} = \mathcal{L}_{box} + \lambda_2\mathcal{L}_{noun} + \lambda_3\mathcal{L}_{verb} + \lambda_4\mathcal{L}_{ttc} + \mathcal{L}_{feat} \quad (7)$$

where $\lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$ are hyperparameters.

4. Experiments

4.1. Datasets

We used the following datasets to validate the effectiveness of our method quantitatively and qualitatively.

Ego4D [15] is currently the largest first-person dataset available, consisting of 5 splits covering distinct tasks and a total of 3,670 hours of videos across 74 different locations. For the next-active-object prediction and STA task, we use the "forecasting split" which contains over 1000 videos and is annotated at 30 fps for the STA task. The dataset annotations include the next-active-objects in the last observed frame, which is a unique feature of that dataset *wrt.* STA. Our goal is to predict the noun class ($\hat{n}$), bounding box ($\hat{b}$), the verb depicting the future action ($\hat{v}$), and the Time to Contact (TTC) (δ) for a given video clip. For comparison on the Ego4D dataset, we apply the existing methods, which are designed for action anticipation-based tasks, by confining them to only predict the next-active-object class label ($\hat{n}$), the verb depicting the future action ($\hat{v}$), and the TTC (δ) since the compared methods are not designed to predict bounding boxes.

Epic-Kitchens-100 [4] consists of about 100 hours of recordings with over 20M frames comprising daily activities in kitchens, recorded with 37 participants. It includes 90K action segments, labeled with 97 verbs and 300 nouns (i.e. manipulated objects). Since the dataset does not provide annotation for next-active-object, we exploit the object detector provided by [5] and also the annotations provided in [37] to curate labelings composed of bounding boxes *i.e;* locations of next-active-objects in the last observed frame. It is to be noted that, to adapt this dataset for the next-active-object detection task, it is imperative that the object, which is used in future action is visible in the last observed frame. However, based on our annotations we realized that for 12.5 % of training data in EK-100 the next-active-object annotations are absent *i.e;* the future active object is not visible in the last observed frame.

4.2. Implementation Details

In order to pre-process the input video clips, we randomly scale the height of the clips between 248 and 280 pixels and take 224-pixel crops for training. We sample 16 frames at 4 frames per second (FPS). We adopt the network architecture of Swin-T [26, 27] to serve as the backbone of our network to extract the video features from the sampled clip. However, we only utilize the outputs till the first-three block of the video swin transformer [25] along with down-sampling of each block to extract the *per-frame* feature maps, which are required later to predict the bounding boxes. We also use a 3-layer multi-head transformer encoder and decoder, which operates on a fixed 256-D. We train our end-to-end model with SGD optimizer using a learning rate of $1e-4$ and a weight decay of $1e-6$ for 50 epochs.

4.3. Evaluation Metrics

We evaluate our models on the Ego4D [15] dataset using the evaluation metrics defined by the dataset creators for short-term anticipation tasks. These metrics include the Average Precision of four different combinations of the next-active-object-related predictions: noun class ($\hat{n}$), bounding box ($\hat{b}$), future action ($\hat{v}$), and time to contact (δ). We use the top-1 accuracy to evaluate the performance of the future action ($\hat{v}$) and next-active-object label ($\hat{n}$) predictions. For bounding boxes ($\hat{b}$) and time to contact (δ), the predictions are considered correct if the predicted boxes have an Intersection over Union (IoU) value greater than or equal to 0.5 and the absolute difference between the predicted and ground-truth time to contact is less than or equal to 0.25 seconds ($|\hat{y}_{ttc} - y_{ttc}| \leq 0.25$). In the case of combined pre-

Models	$AP_{\hat{b}}$	$AP_{\hat{b}+\hat{n}}$	$AP_{\hat{b}+\hat{n}+\delta}$	$AP_{\hat{b}+\hat{n}+\hat{v}}$	$AP_{\hat{b}+\hat{n}+\hat{v}+\delta}$	$AP_{\hat{b}+\delta}$	$AP_{\hat{b}+\hat{v}}$	$AP_{\hat{b}+\hat{v}+\delta}$
Slowfast [15]	40.5	24.5	5.0	4.9	1.5	8.4	8.16	1.9
Slowfast (with Transformer backbone)	40.5	24.5	4.5	4.37	1.73	7.5	8.2	1.3
AVT [13]	40.5	24.5	4.39	4.52	1.71	7.12	8.45	1.15
ANACTO [13]	40.5	24.5	4.55	5.1	1.91	7.47	8.9	1.54
MeMVIT [39]	40.5	24.5	4.95	5.89	1.34	9.27	10.04	2.11
Ours	**45.3**	**27.0**	**9.0**	**6.54**	**2.47**	**16.6**	**12.2**	**4.18**

Table 1. Results of our model and other baseline methods on Ego4D [15] dataset for different output targets, bounding box ($\hat{b}$), next-active-object label ($\hat{n}$), future action ($\hat{v}$) and the time to contact with the object (δ) based on their Average Precision (AP).

Model	Params (M)	Init	Unseen			Tail			Overall		
			Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun
Chance	-	-	0.5	14.4	2.9	0.1	1.6	0.2	0.2	6.4	2.0
TempAgg (RGB) [36]	-	[6]	12.2	27.0	23.0	10.4	16.2	22.9	13.0	24.2	29.8
RULSTM [12]	-	[35]	-	-	-	-	-	-	7.8	17.9	23.3
RULSTM [12]	-	[6]	13.1	28.8	23.7	10.6	19.8	21.4	13.25	27.5	29.0
AVT (RGB) [13]	393	[6]	-	-	-	-	-	-	14.9	30.2	31.7
AVT + [13]	-	[6]	11.9	**29.5**	23.9	**14.1**	21.1	25.8	**15.9**	28.2	32.0
MeMViT [39]	59	[21]	9.8	27.5	21.7	13.2	**26.3**	**27.4**	15.1	**32.8**	**33.2**
Ours	23.5	-	**14.3**	29.3	**27.8**	4.4	13.2	13.8	10.7	25.3	27.9

Table 2. Results of our model and other baseline methods on EK-100 [4] dataset on validation set. "Overall" comprises of samples combining the Unseen and Tail set plus also consisting of *seen* samples from the training set.

dictions involving two or more unknowns, the prediction is deemed correct only if all the unknowns are predicted correctly. For the purpose of training, we kept the values of all λ as 1, except λ_4 which is set to 10 following [15].

For comparison of models on the EK-100 [4] dataset, we adhere to the metric commonly used in recent action-anticipation works [12, 13, 39].

4.4. Comparison with State-of-the-art

For Ego4D dataset [15], we compare our model with the methods restricted to only predicting the future action ($\hat{v}$) and TTC (δ) of a given sample clip, since the only methods we perform a comparison with, are action anticipation methods that have not been designed to predict bounding boxes. Table 1 declares the results for Ego4D [15] dataset. We observe that our model achieves better performance than the object detector [35] that is pre-trained on Ego4D, in terms of predicting the NAO's class label and bounding box location, as evidenced by the higher $AP_{\hat{b}}$ and $AP_{\hat{b}+\hat{n}}$ scores. This superiority is also visually evident in Fig. 3, where the performance of our object decoder is shown to refine the detected objects for NAO and even identify objects that were not detected by the object detector. Moreover, our model outperforms all the other baseline methods across all other evaluation metrics for the STA task.

In the case of EpicKitchen-100 dataset [4], we compare our proposed method against SOTA for **action anticipation task**, as described in [5, 12]. It is important to note that *the action anticipation task differs significantly from the STA task*, where the concept of next-active-object is not considered. However, we compute our own annotations to adapt the Action Anticipation task for STA-based scenarios, as discussed in Sec. 4.1. Since our model and the STA task require the identification of the next-active-object (and its visibility/presence) in the last observed frame, this is reflected in our results due to the limitations of the dataset. The results of our experiments on the EK-100 dataset are presented in Table 2. We achieve state-of-the-art performance on the "Unseen Set" which only contains a small fraction (6%) of samples where no Next-Active-Object (NAO) is detected in the last observed frame. It is to be noted that NAO-GAT is the lightest *w.r.t.* other compared models. However, our model's performance on the "Tail Set" is suboptimal, likely due to the fact that the NAO is not visible in the last observed frame for around 22% of the clips. This limitation causes confusion in our model, which relies on the visibility of NAO in the last frame, and impacts the overall results for the "Overall Set," which comprises the "Unseen Set", "Tail Set," and training set's "seen" samples.

To investigate the impact of the "Tail Set" on the "Overall Set" accuracy, we remove clips corresponding to tail classes for which NAO is not present in last observed frame and observe improvements of $+5.2 \uparrow (16.9\%)$, $+4.0 \uparrow (32.4\%)$, and $+7.6 \uparrow (35.5\%)$ in action, verb, and noun recognition, respectively.

We report additional qualitative results of our model on both dataset in our supplementary material.

4.5. Ablation study

We conducted an ablation study on Ego4D dataset to analyze the impact of different modules of the proposed method in Table 3. We evaluated the performance of our

Model	$AP_{\hat{b}}$	$AP_{\hat{b}+\hat{n}}$	$AP_{\hat{b}+\hat{n}+\delta}$	$AP_{\hat{b}+\hat{n}+\hat{v}}$	$AP_{\hat{b}+\hat{n}+\hat{v}+\delta}$	$AP_{\hat{b}+\delta}$	$AP_{\hat{b}+\hat{v}}$	$AP_{\hat{b}+\hat{v}+\delta}$
Ours w/o *OMD*, *OD*	45.3	26.7	4.78	5.55	1.05	7.89	8.91	1.52
Ours w/o *OMD*	45.1	27.1	4.48	6.2	1.0	7.34	10.2	1.44
Ours (ResNet50)	42.7	25.2	4.3	6.0	1.1	10.6	10.1	2.0
Ours (Full)	**45.3**	**27.0**	**9.0**	**6.54**	**2.47**	**16.6**	**12.2**	**4.18**

Table 3. Ablation study performed on ego4D [15] to investigate the effect of Backbone, Motion Dynamics (MD), and object decoder (OD) modules on the motion-based output sequences by the model.

Figure 3. The top row (a) shows the "last observed frame" and all the object detections provided by the object detector [35]. The bottom row (b) depicts the output from our motion decoder. It can be observed that our model learns from past observations and selects the best possible object(s) for the next-active-object selection in the frame. Besides, it can be seen that it is even able to identify objects which were not detected by the object detector (3^{rd} and 7^{th} column are the clearest examples).

complete model in comparison to the models that omit either the Object Decoder (OD) module (Section 3.4) or the Object Motion Dynamics (OMD) module (Section 3.5 along with Object Decoder (OD) together. Our findings indicate that the Object Decoder module improves the prediction of future verbs ($\hat{v}$), resulting in a higher $AP_{\hat{b}+\hat{v}}$. This suggests that having prior knowledge of the future active-object can support anticipating future action. On the other hand, the OMD module plays a crucial role in estimating the time needed to make contact with the next-active-object and initiate an action, resulting in a significant improvement in $AP_{\hat{b}+\delta}$ and other related metrics. Since OMD provides additional background motion information which helps the model greatly in predicting the TTC. These findings suggest that both modules are essential for accurately anticipating future actions in first-person videos. Additionally, we also investigated the impact of the backbone network on our model's performance. For this purpose, we replace our Swin-T backbone with ResNet50 [16] architecture. Using ResNet50 demonstrates a significant drop in performance across all metrics.

5. Conclusion

We have investigated the problem of short-term action anticipation using the next-active-objects. First, we discussed the formulation of the STA task. We then presented a new vision transformer-based model, which learns to encode human-object interactions with the help of an object detector and decode the next-active-object location in the last observed frame. We then demonstrated the importance of next-active-object information to predict the future action and time to start the action using additional background information as object motion dynamics. We proved the proposed method's effectiveness by comparing it against relevant strong anticipation-based baseline methods. In future work, we will investigate the use of an object tracker with other human-centered cues such as gaze and the appearance of objects over time. We will also investigate the effect of action recognition on *NAO* identification and localization.

Limitations. As discussed above, the proposed approach is specifically designed for Short-Term Anticipation (STA) task, where the next-active-object is assumed to be visible in the last observed frame. Therefore, when applied to a slightly different task of Action Anticipation, our model shows limitations as it relies on this assumption which does not necessarily hold true in this case.

Broad Impact. The proposed method can be used in several real-world applications such as in robotics or virtual / augmented reality. In case the first person also interacts with other people but not only the non-living objects, then there might be issues regarding privacy preservation. In such cases, policy reviews should be further considered when using the proposed method.

6. Acknowledgement

This research is supported by the project Future Artificial Intelligence Research (FAIR) – PNRR MUR Cod. PE0000013 - CUP: E63C22001940006.

References

[1] Anna M Borghi. Object concepts and action. *Grounding cognition: The role of perception and action in memory, language, and thinking*, pages 8–34, 2005.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *International Journal of Computer Vision*, 2021.

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] Eadom Dessalene, Chinmaya Devaraj, Michael Maynord, Cornelia Fermuller, and Yiannis Aloimonos. Forecasting action through contact representations from first person video. *IEEE TPAMI*, pages 1–1, 2021.

[8] Eadom Dessalene, Michael Maynord, Chinmaya Devaraj, Cornelia Fermuller, and Yiannis Aloimonos. Egocentric object manipulation graphs. *arXiv preprint arXiv:2006.03201*, 2020.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[11] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation*, 49:401–411, 2017.

[12] 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.

[13] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *ICCV*, 2021.

[14] Ross Girshick. Fast r-cnn. In *IEEE ICCV*, pages 1440–1448, 2015.

[15] Kristen Grauman, Andrew Westbury, and Eugene et al. Byrne. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] Julia Hertel, Sukran Karaosmanoglu, Susanne Schmidt, Julia Bräker, Martin Semmann, and Frank Steinicke. A taxonomy of interaction techniques for immersive augmented reality based on an iterative literature review. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 431–440, 2021.

[18] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3148–3159, June 2022.

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] Jingjing Jiang, Zhixiong Nan, Hui Chen, Shitao Chen, and Nanning Zheng. Predicting short-term next-active-object through visual attention and hand position. *Neurocomputing*, 433:212–222, 2021.

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[22] Kyungjun Lee, Abhinav Shrivastava, and Hernisa Kacorri. Leveraging hand-object interactions in assistive egocentric vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[23] Miao Liu, Siyu Tang, Yin Li, and James Rehg. Forecasting human object interaction: Joint prediction of motor attention and actions in first person video. In *ECCV*, 2020.

[24] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin trans-

former: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[27] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[28] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211, June 2022.

[29] Blascovich J.J. Loomis J.M. and A.C. Beall. Immersive virtual environment technology as a basic research tool in psychology. *Behavior Research Methods, Instruments, and Computers*, 31:557–564, 1999.

[30] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1046–1056, 2020.

[31] Tushar Nagarajan and Kristen Grauman. Shaping embodied agent behavior with activity-context priors from egocentric video. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29794–29805. Curran Associates, Inc., 2021.

[32] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE CVPR*, pages 2847–2854, 2012.

[33] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[34] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1569–1578, January 2021.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[36] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020.

[37] Sanket Thakur, Cigdem Beyan, Pietro Morerio, Vittorio Murino, and Alessio Del Bue. Anticipating next active objects for egocentric videos, 2023.

[38] Xiaohan Wang, Linchao Zhu, Heng Wang, and Yi Yang. Interactive prototype learning for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8168–8177, October 2021.

[39] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition. In *CVPR*, 2022.

[40] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *IEEE Transactions on Image Processing*, 30:1143–1152, 2020.

[41] Zeyun Zhong, David Schneider, Michael Voit, Rainer Stiefelhagen, and Jürgen Beyerer. Anticipative feature fusion transformer for multi-modal action anticipation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6068–6077, January 2023.