On the Efficacy of Text-Based Input Modalities for Action Anticipation

BMVC 2024 Submission # 560

Abstract

Anticipating future actions is a highly challenging task due to the diversity and scale of potential future actions; yet, information from different modalities help narrow down plausible action choices. Each modality can provide diverse and often complementary context for the model to learn from. While previous multi-modal methods leverage information from modalities such as video and audio, we primarily explore how text descriptions of actions and objects can also lead to more accurate action anticipation by providing additional contextual cues, e.g., about the environment and its contents. We propose a Multi-modal Contrastive Anticipative Transformer (M-CAT), a video transformer architecture that jointly learns from multi-modal features and text descriptions of actions and objects. We train our model in two-stages, where the model first learns to align video clips with descriptions of future actions, and is subsequently fine-tuned to predict future actions. Compared to existing methods, M-CAT has the advantage of learning additional context from two types of text inputs: rich descriptions of future actions during pre-training, and, text descriptions for detected objects and actions during modality feature fusion. Through extensive experimental evaluation, we demonstrate that our model outperforms previous methods on the EpicKitchen datasets, and show that using simple text descriptions of actions and objects aid in more effective action anticipation. In addition, we examine the impact of object and action information obtained via text, and perform extensive ablations. We will release code upon acceptance.

1 Introduction

Suppose you go to a cafe and order a coffee and you see your barista steaming milk, can you predict what they might do next? Action anticipation is the task of predicting future actions, using visual cues and data from other modalities such as audio, sensor data, etc. from current and prior observations. Predicting future actions is important for many Artificial Intelligence (AI) applications such as autonomous driving [18, 34], assistive robotics [22, 23, 31], augmented reality, etc. Although seemingly straightforward for humans, this task is difficult for AI models due to the challenging nature of predicting the future and the wide range of possible actions that the models have to learn. Models not only have to detect the action happening at the observed time, but also fuse information from (all available) modalities to anticipate future actions.

Anticipation using only videos (single modality) remains challenging and the availability of additional and complementary modalities is typically advantageous [13, 49]. For instance, an assistive robot can be prepared to help an elderly person if the robot can detect the events leading up to a fall, and anticipate it. In addition to video (camera data), audio (sound of the fall, or the person's scream), a third person's audio command ("Help the person") etc are beneficial. Accordingly, recent works [13, 39, 49] have shown that action anticipation greatly benefits from multi-modal training, e.g., using visual and audio cues such as active object detection, and hand-object contact information, ASR etc. Although the models are typically trained using modality specific encoders, we examine *if natural language descriptions of actions and objects can be useful for action anticipation*, when employed in addition to other modalities. Such descriptions can be highly useful as they can incorporate additional context about the environment and the objects required for performing the actions, e.g., kitchen vs living room, the utensils utilized, etc., leading to improved action anticipation. To this end, we leverage the in-context learning capabilities of Large Language Models (LLMs) to generate rich and detailed descriptions of actions and objects.

In this paper, we present a 'Multi-Modal Contrastive Anticipative Transformer (M-CAT)', that employs a two-stage training process: *(i) contrastive pre-training:* where embeddings from videos and other modalities such as optical flow, audio, natural language descriptions of objects and actions are fused, and contrasted against rich text descriptions of *future actions*; and *(ii) fine-tuning:* where the learned embeddings from the modalities are once again fused, and a classifier is trained to predict future actions. For both stages, we utilize frozen pre-trained language models (e.g., the CLIP text encoder) to obtain embeddings for text descriptions of object and actions, in lieu of relying on traditional feature extraction methods.

We study which modalities are more beneficial for action anticipation, and inspect how the accuracy of action recognition for the observed frames affects anticipation. As contrastive pre-training typically requires large batch sizes, we explore alternate avenues of adding more samples during training, specifically for resource constrained setups. Finally, we also investigate whether the utilization of self-supervision as an additional objective can be useful for anticipating actions. Therefore, the contributions of our work are:

- We propose a novel approach for predictive video modeling by contrasting multi-modal features against rich text descriptions for future actions, generated using LLMs.
- We investigate whether natural language descriptions of actions and objects can result in improved action anticipation.
- We improve contrastive pre-training for small batch size capabilities and also introduce an additional self-supervised learning objective.

2 Related Work

Action Anticipation is the task of predicting future actions after certain time units in a given video clip. This task has been explored extensively for third-person videos [1, 12, 17, 19]. The release of large-scale egocentric datasets and challenges such as Epic-Kitchen [6, 7] and Ego-4D [16] have fast tracked the development for first-person scenarios as well. To model the temporal progression of past actions, [11] used a rolling-unrolling-based LSTM network to anticipate actions, such that rolling LSTMs account for the observed video frames, while unrolling LSTMs accounted for the anticipation. [37, 38] made use of long-range past information by building a multi-scale temporal aggregating framework. [39, 40] localize the next active object's position to anticipate actions. In addition to gathering strong visual features, recent methods have used other visual cues like modeling the environment [30]

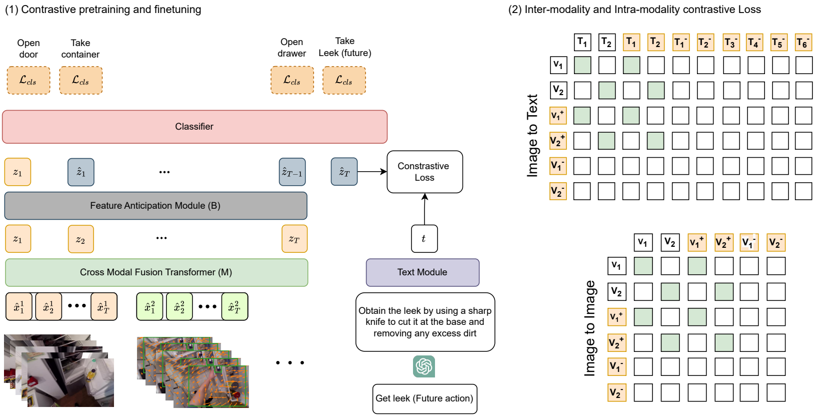

Figure 1: **Left:** Our training comprises two stages: first, contrastive pre-training, where we fuse embeddings from different modalities using a Fusion Module $\mathcal{F}$, followed by an anticipation module $\mathcal{B}$. The output is contrasted against the rich descriptions of future actions generated using a LLM. The second stage involves fine-tuning a linear layer to predict future actions. **Right:** Illustration of the image-text and image-image contrastive setup.

or hand-object contact and activity modeling [8]. More recently, the use of vision transformers [9] has also been explored. While, AVT [13] proposes causal modeling of video frames, and using self-supervision to learn the future frame features, MeMViT [42] perform multi-scale representation of frame features by hierarchically attending the previously cached “memories". AFFT [49] proposes a fusion method to effectively fuse features from multiple modalities and extend AVT for action anticipation. [36], AntGPT [48] and leverages the goal information to reduce the uncertainty in future predictions. AntGPT [48] trains Large Language Models (LLM) to infer goals and model temporal dynamics. In contrast, we use pretrained LLMs to generate additional contextual cues about the actions, and create additional text based modalities from objects and actions.

Language-Image Pre-training Training images jointly with natural language text (e.g., captions) has been established as an effective pre-training method for zero-shot learning, open vocabulary testing, and as well as classification tasks. CLIP [33], ALIGN [20], FLorence [46], X-CLIP [27], UniCL [44] have shown that contrastive training on large-scale image-text pairs results in astonishing performance for zero-shot prediction. OWL-ViT [28] uses a CLIP-based contrastive approach to transfer image-level pre-training to open vocabulary object detection. Similarly, CoCa [45] is not only trained on the contrastive loss, but also leverages generative modeling via a captioning loss. Flamingo [2] on the other hand interleaves visual data with text and produces free-form text as output, demonstrating effective performance on several downstream tasks. Such natural language supervision also aids in video representation learning. For instance, [3] used a visual detector to map every object instance in the video frame into its contextualized word representation obtained from narration. Building on these works, we propose a CLIP-like contrastive pre-training approach that learns to align multi-modal features with rich descriptions of future actions.

Multi-modal training Typically, modalities used for action anticipation include RGB images, optical flow, object information, IMU, and audio [10, 13, 37, 43, 47, 49]. Features from each modality are averaged, either weighted [13] or unweighted [10], or a Multi-Layer Perceptron (MLP) is used [21]. Recently, multi-head cross attention is being employed to attend over different modalities [25, 49]. However, training modality specific encoders can be computationally expensive. Instead, we explore the usage of text based inputs as modalities ie objects and actions detected in text form in lieu of visual features. To this end, we propose an architecture which contrasts fused features from different modalities including text from actions and objects detected in the video, with descriptions generated from action labels.

3 Methodology

Given a video segment starting at τ_s, the goal is to anticipate action using τ_o length of observed segment τ_a units before it, *i.e.* from $\tau_s - (\tau_a + \tau_o)$ to $\tau_s - \tau_a$. The anticipation time τ_a is usually fixed for each dataset, while the observation time τ_o can be varied. We extract T temporally sequential inputs for M modalities, denoted as x_i^m, $i \in \{1, \dots, T\}$ and $m \in \{1, \dots, M\}$. Please refer to the Appendix for an illustration of the action anticipation task.

Our model architecture (shown in Figure 1) comprises two stages: contrastive pre-training and fine-tuning to perform action anticipation. During pre-training, the model consists of M modality specific feature extractors $\mathcal{B}_m$, $m \in \{1, \dots, M\}$, a fusion model $\mathcal{F}$, and an anticipative module $\mathcal{B}$. In the fine-tuning stage, an additional classifier is trained to predict the future action, while the rest of the model is kept frozen. We utilize the fusion module from [49], and a variation of the GPT2 model used in [13] for feature anticipation to predict $\hat{z}_{i+1} = \mathcal{D}(z_i), i \in \{1, \dots, T\}$. In what follows, we detail the two stages, along with the implementation details. Throughout, all modality feature are extracted from pre-trained models.

3.1 Pre-training

We employ a CLIP-like [33] setup, where the embeddings from different modalities (e.g., images and audio) are contrasted against text embeddings computed from text descriptions of future action classes (detailed below). The setup utilizes the following modality features:

Video Features: Given a video segment V consisting of T frames, the backbone network B extracts features for each frame. Following [49], we use the Swin transformer features extracted with Omnivore [14], which was trained for action recognition.

Other Modality features: For other modalities like audio, optical flow, etc., we use the features provided by the official repositories [11, 49].

Text Embeddings for Descriptions of Actions and Objects: The embeddings for text data are extracted using a pre-trained CLIP text encoder, which is kept frozen during training, and only a modality specific projection layer is trained. The setup for obtaining the text descriptions for actions and objects is the following: *(i) objects in the video:* the objects present in the (current) video are detected using a pre-trained FasterRCNN model [11]. They are converted into a sentence using the template: `A video containing the following objects: <list of objects>`, and encoded using the aforementioned CLIP text encoder; and *(ii) actions in the video:* similarly, we also generate a sentence for the (current) actions in the video using the template: `A video containing the following actions: <list of action>`. As some datasets do not have dense action annotations, whenever actions are not available, we use the "no action" tag. During both pre-training and action anticipation, we use ground-truth action labels. However, we also analyze the impact of the action recognition accuracy on action anticipation in Section 4.3.

Cross Modal Fusion: For fusing information from multiple modalities x_i^m, we use the self-attention fuser (SA-Fuser) blocks from [49]. It applies L consecutive Transformer en-

Dataset	τ_a	Modalities	Metrics
EGTEA+	0.5s	RGB, Flow	Top-1, cm Top-1
Ek55	1.0s	RGB(R), Obj(O), Flow(F), Audio(A), Objects (text)(U), Actions(text)(V)	Top-1, Top-5
EK100	1.0s	RGB(R), Obj(O), Flow(F), Audio(A), Objects (text)(U), Actions(text)(V)	Recall@5

Table 1: Modalities and metrics used for different datasets.

coders at each time step with dimentionality of d and k attention heads, and contains a learnable token x^{Λ}. The final output is the mean of all learnable tokens.

Anticipation: The Fused embeddings are passed through a variation of the GPT-2 [32] module to predict the future features: $\hat{\mathbf{z}}_1, \ldots \hat{\mathbf{z}}_T = \mathcal{D}(\mathbf{z}_1, \ldots, \mathbf{z}_T)$ where $\hat{\mathbf{z}}_t$ is the predicted feature corresponding to the frame $\mathbf{z}_t$ after attending to the frames $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$. We refer the reader to [13] for more details.

Text Emeddings for Rich Descriptions of Future Actions: we generate diverse and context rich text descriptions of the action classes using GPT3.5 [4] (from the OpenAI API), by converting the class names into sentences using the prompt: `Describe <xyz> action in 1 sentence in 10 different ways`, and randomly select one response during training. For reference, we provide examples and details about this generation in the Appendix.

Pre-training: Features at z_T, which have encoded the temporal information over all observed frames, are then trained to align with the text embedding for the future action via contrastive learning. As the models were trained using smaller batch sizes, for effective contrastive learning, we augment the training with additional positive and negative samples. As our model is trained on features and not raw videos, instead of applying augmentations to the videos to genereate more positive samples, we follow a slow-and-fast approach. For every sample (fast) in the batch, we create another positive sample (slow) where we uniformly sample $(1/4)T$ number of frames (denoted as V_i^+ in Figure 1b), and randomly shuffle the temporal order for negative samples (V_i^-). In addition, we contrast every video samples against all other action classes that do not appear in the batch (T_i^-). In order to limit the memory usage, we cap the number of negative text samples to 512. So, for a input batch-size of 128, we have a total of $N_v = 128 * 3$ samples for videos, and $N_t = (128*3)*(128*2+512)$ samples for text (see Figure 1b), with increase of 280k samples, from $\sim$ 16k to $\sim$ 300k. In every iteration, the model has 0.0008:1 ratio of positive to negative samples, close to using a batch-size of 1024, as opposed to 0.008 when a batch-size of 128 is used.

Similar to SLIP [29], we also add a self-supervised learning objective. The positive and the negative samples curated (V_i^+ and V_i^-), along with original samples (V_i) are trained such that similar samples are pushed closer in the embedding space.

We use standard cross entropy to train the contrastive loss. The loss is defined as

$$\mathcal{L}_{cross} = (\mathcal{L}_{v2t} + \mathcal{L}_{t2v}) * 0.5 + \mathcal{L}_{v2v} \tag{1}$$

Following AVT [13], we also utilize a self-supervised feature loss $\mathcal{L}_{feat}$ and L_{next} in addition to the contrastive loss. Therefore, our final loss function is $\mathcal{L} = \mathcal{L}_{cross} + \mathcal{L}_{feat} + \mathcal{L}_{next}$, where $\mathcal{L}_{feat}$ is defined as mean squared error between $\hat{\mathbf{z}}_t$ and $\mathbf{z}_{t+1}$, which matches the future features predicted with the true features in a self-supervised manner.

3.2 Fine-tuning Network

Here, we fine-tune the classifier layers for the action anticipation task. We use the features obtained from the feature anticipation module, $\hat{z_T}$, in conjunction with a linear layer, and train with the cross entropy loss $\mathcal{L}_{cls}$. During the fine-tuning stage, the fusion ($\mathcal{M}$) and the anticipation module ($\mathcal{B}$) are kept frozen when the same modalities are used, and the fuser is finetuned when different modalities are used during pre-training and fune-tuning.

3.3 Implementation details

We process the input videos similar to [13], and sample 16 frames at 1 fps, by setting $\tau_o = 16s$. We use the Swin Transformer based RGB features provided by [49], which were extracted from the Omnivore [14] network, originally trained for action recognition. We use the pre-trained CLIP text encoder, processor, and tokenizer, provided by [41] for processing all text inputs. During pre-training, the encoded features are projected to 1024 dimensions, before passing through the fusion and the anticipative modules. In the fine-tuning stage, the fused features are classified using a single linear layer. For both stages, we use the SGD+momentum optimizer, using a learning rate $1e^{-3}$ and weight decay $1e^{-6}$ for 50 epochs. Further, we employ a cosine annealing learning rate schedule with a warmup for 20 epochs, and the training is performed on a single Nvidia A40 GPU, with a batchsize of 128.

For the optical flow and object features, we use the official RULSTM [11] repository, and for audio, we use features provided by [49] [1]. Our code, weights, and the action descriptions generated will be publicly released upon acceptance.

4 Experiments

4.1 Experimental setup

Datasets and metrics: We evaluate on three action anticipation datasets: *(i)* Epic-Kitchens100 (EK100) [7], reporting the class-mean Recall@5 for actions, verbs and nouns; *(ii)* EpicKitchens55 (EK55) [6], where we report the Top-1 and Top-5 for actions, verbs and nouns, through standard train and val splits; and *(iii)* EGTEA Gaze+ [24], in which we report the performance on the first split of the dataset at τ_a = 0.5s, and the metrics include Top-1 and class-mean(cm) Top-1 accuracies for actions, nouns and verbs. We add further details about these datasets in the Appendix.
Modalities: We summarize the modalities and metrics used in Table 1. We use pre-trained TSN weights provided by the official repositories [11, 49] for object features, audio, and flow. We use the objects detected using the FasterRCNN model trained on Epic-Kitchen 55 dataset [11], and use a threshold of 0.15 and pick the top 5 objects for every frame in the video. For actions, we use the labels provided by the dataset during training and evaluation. We evaluate the impact of action recognition accuracy and discuss the results in Section 4.3.
Baselines: We evaluate our approach against the state-of-the-art for action anticipation, including, RULSTM [11], AVT [13], ActionBanks [37], AFFT [49], and MeMViT [42] [2]. We re-train the AFFT model on our local environment setup for fair comparison, and observe a small discrepancy in performance relative to the published paper. As the goal of this paper is to demonstrate the effectiveness of learning from text embeddings, we do not compare against other state-of-the-art methods that have a substantially different architectures like [35, 36, 48]. AntGPT [48] introduces a promising alternate way of predicting future actions by fine-tuning LLMs. Yet, we do not compare against it, as we do not use LLMs to infer our outputs or predict goals and actions, rather only use it to generate detailed descriptions.

[1] Please refer to [49] for details about the feature extraction for different modalities.
[2] Please see the Appendix for details about the baselines.

Method	Verb		Noun		Action	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
RULSTM	32.4	79.6	23.5	51.8	15.3	35.3
ActionBanks	**35.8**	80.0	23.4	52.8	15.1	35.6
AVT	-	-	-	-	14.4	31.7
AVT+	32.5	79.9	24.4	54	16.6	37.6
AFFT	34.9	78.7	26.2	53.9	17.0	34.3
Ours (R -> R)	32.4	80.1	28	56.4	16	36.5
Ours(ROFA -> ROFA)	33	79.4	26	55.5	14.9	35.9
Ours(R -> ROFA)	32.5	80.4	27.8	57	16.5	38.1
Ours* (R -> ROFA+UV)	34.3	**80.6**	**29.7**	**58.8**	**17.9**	**39.8**

(a) EK55

Model	Top-1			Class mean acc		
	Verb	Noun	Act.	Verb	Noun	Act.
I3D-Res50 [5]	48.0	42.1	34.8	31.3	30.0	23.2
FHOI [26]	49.0	45.5	36.6	32.5	32.7	25.3
AVT(TSN) [13]	51.7	50.3	39.8	**41.2**	41.4	28.3
AFFT [49]	**52.1**	**50.7**	**41.4**	38.4	**43.7**	**31.8**
Ours (R -> RF)	51.4	49.7	40.8	38.9	43.3	31.3

(b) EGTEA Gaze+

Table 2: **(a) EK55:** Comparison of state-of-the-art methods on the validation set of EK55 using the modalities (ROFA). * indicates that additional action (V) and objects (U) information was provided in the text form. R indicates that only RGB features were used, ROFA refers to RGB, Obj(TSN), flow (TSN) and Audio features. **(b) EGTEA Gaze+:** Model performance for Split=1 at $\tau_a = 0.5s$. **Bolded** values indicate highest score, and -> denotes the modalities used for pre-training and fine-tuning.

Additionally, they report performance on few-shot for the Ego4D dataset (which we did not evaluate on) making one-to-one comparison hard.

ChatGPT generated action descriptions: We provide examples of the descriptions generated for actions using the ChatGPT API (with GPT3.5 Turbo) in the Appendix. In the descriptions, there are generally mentions of other objects that are used when the action takes place. For example, the text descriptions for "take chopsticks" are *"Use chopsticks to grasp food and bring it to your mouth"*, *"Take the chopsticks and use them to pick up the food"* etc, giving context about other objects in contact with hand etc. Similarly, descriptions for the "mix mushroom" action often involve words such as tongs, spoons or a spatula.

4.2 Comparison Against Baselines

EGTEA+ In Table 2b, we compare our results on split 1 (as in [26]) at $\tau_a = 0.5s$. In addition to the RGB data, we use the flow data provided by [11]. Similar to AFFT, we use the pre-trained TSN features. We also note that the results for AFFT were obtained by using the official code on our local environment. We observe that our approach does not improve performance on EGTEA+, in contrast to other larger datasets. The smaller scale of EGTEA+ is not a good match for contrastive learning, which is generally sensitive to data size and sample variety, and thereby, does not result in performance improvement.

Epic-Kitchen In Table 2a and Table 3, we compare the performance of our method to the state-of-the-art for the EK55 and EK100 datasets. For EK55, we obtain the results for the AFFT baseline using the authors' code. First, we consider the performance of our approach when trained using only the (single) RGB modality. As the fusion module is a block of transformer layers, they act as feature encoder layers when there no modalities to fuse. We observe that our method has a 2% absolute improvement over AFFT, a multi-modal method, and ~5% to AVT's single modality performance. When the model is pre-trained and fine-tuned with multiple modalities (ROFA->ROFA), it outperforms AFFT for the Top-5 metrics, yet performs poorly compared pre-training solely on RGB. This leads us to evaluate pre-training with just the RGB modality, while fine-tuning the fusion and classifier layers with multiple modalities (R->ROFA). With this training strategy, we see an improvement of 2% compared to the (ROFA->ROFA) training, and outperforming our single modality perfor-

Method	Overall			Unseen			Tail		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
RULSTM	27.8	30.8	14.0	28.8	27.2	14.2	19.8	22.0	11.1
TempAgg	23.2	31.4	14.7	28	26.2	14.5	14.5	22.5	11.8
AVT	30.2	31.7	14.9	-	-	-	-	-	-
AVT+	28.2	32.0	15.9	29.5	23.9	11.9	21.2	25.8	14.1
MeMViT	**32.3**	37.0	17.7	28.6	27.4	15.2	25.3	31.0	15.5
AFFT(Swin+)	22.8	34.6	18.5	24.8	26.4	15.5	15.0	27.7	16.2
AFFT (re)	22.4	32.4	18.1	26.5	26.8	15.3	14.6	24.3	15.9
Ours (R -> R)	30.1	32	16	32.7	28.4	15.3	23.4	25.3	13.8
Ours (R->ROFA)	31.9	35.9	17.3	32.5	30.2	14.5	**25.9**	30.3	15.4
Ours* (R -> ROFA+UV)	31.3	**47.8**	**23.8**	**34.5**	**42.8**	**24**	23.8	**41.9**	**20.3**

Table 3: **EK100**: comparison of state-of-the-art method on the validation set of EK100 using modalities provided by [ ]. MeMViT uses only RGB data, while the rest use multiple modalities. R indicates that only RGB features were used, ROFA refers to RGB, Obj(TSN), flow(TSN) and Audio features. * indicates that additional action (V) and object(U) modalities in the text form were used. **Bolded** values indicate the best performing method, and -> denotes the modalities used for pre-training and fine-tuning.

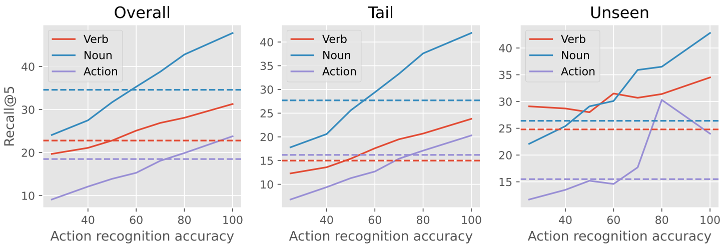

Figure 2: Impact of action recognition accuracy on the prediction of verbs, nouns and actions for EK100. Values in dashed lines are the corresponding results from the AFFT baseline.

mance by 0.5% for Top-1 and 1.3% for Top-5 metric for actions. We believe that during pre-training, the model can find it challenging to fuse all the modality features and then align them with text embeddings. Additionally, we hypothesize that an Imagebind-like training setup might be beneficial, however, we do not evaluate this scenario as ImageBind trains modality specific encoders, whereas we do not. Adding additional information about the objects and actions, we see an absolute improvement of 1% for Top1, and 5% for Top5.

For EK100, we compare our two-stage network against single-stage methods, as well as using action and object information in the text form in Table 3. Similar to EK55, we see that our single modality method outperforms AVT, while performing comparable to MeMViT. However, MeMViT is a method that is directly trained on the videos, while we used pre-extracted features. Our multi-modality method (R->ROFA), while performing similarly to AFFT for actions, shows a signification improvement in the verb and noun predictions. In addition, with the action and object information, we see a clear improvement across all predictions, particularly in the unseen and the tail category. This indicates that the model has efficiently learnt from the additional context provided by the text representations.

4.3 Ablations and Analysis

Impact of modalities: In Table 4b, we explore the contributions of various modalities to EK100's performance. Using RGB as one of the 'modalities', we examine the contributions by audio and actions to the performance. In detail modality contributions are discussed in

Method	Verb		Noun		Action		
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Recall@5
Ours (w/ gpt)	**32.8**	79.8	27.8	56.5	15.6	**36.8**	16.1
Ours (w/o gpt)	31.9	79.6	26.6	56.1	14.8	36.7	16
Ours (w/o Aug)	31.9	79.5	26.6	56.1	14.8	36.3	14.8
Ours (w/ L_{v2v})	32.4	**80.1**	**28**	**56.4**	**16**	36.5	**17.5**

(a) EK55 Ablations

Method	Overall			Unseen			Tail		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Ours (ROFA->ROFA)	28	33.9	15.8	29.4	28	16.6	20.5	27	13.3
Ours (R -> R)	30.1	32	16	32.7	28.4	15.3	23.4	25.3	13.8
Ours (R -> RV)	28.1	44.3	21.8	36.7	41.5	23.3	20.0	37.9	18.6
Ours (R -> RAV)	30.6	44.6	22.4	**41.0**	40.5	21.9	23.1	38.4	19.2
Ours (R -> ROFA+UV)	**31.3**	**47.8**	**23.8**	34.5	**42.8**	**24**	**23.8**	**41.9**	**20.3**

(b) EK100 Ablations

Table 4: **Left: EK55 Ablations**: comparing different training losses and protocols on the validation set of EK55 using only RGB. w/ and w/o gpt indicate pre-training with/without using descriptions of future actions. w/o Aug indicates that slow-fast and negative samples were not appended to the batch samples during the pre-training, while all other methods were not trained using L_{v2v} loss, w/ L_{v2v} is trained with the loss in addition to other losses and data augmentations. **Right: EK100**: Impact of different modalities on model performance.

the Appendix. We see that from the model's performance that action and the audio provide complementary information that RGB alone could not leading to a better performance.

Impact of different training settings: We evaluate the effect produced by the losses and data augmentations used during pre-training in Table 6, where: *(i) w/ gpt* indicates that ChatGPT generated action descriptions were used during pre-training (as detailed in Section 4.1); *(ii) w/o gpt* involves pre-training with the the simple template - `This is a video clip with action <xyz>`; *(iii) w/o Aug* indicates that during training, the batch samples were not appended with positive and negative samples from the slow-fast and randomly shuffled features(detailed in Section 3.1); and *(iv) w/* L_{v2v} contains the self-supervised loss in addition to other losses during pre-training. We observe that using the richer descriptions from ChatGPT and the self-supervised loss L_{v2v} boosts the action prediction performance by 1.2% for Top-1, 0.2% for Top-5, and 3% for recall@5, indicating their necessity during training.

Effect of the Accuracy of Actions: In Figure 2, we evaluate the impact of having access to accurate actions on action anticipation. For this evaluation, we vary the accuracy %-age of ground-truth action labels used. Therefore, when the accuracy of actions is 20%, it indicates that 80% of actions during training are incorrect (i.e., they are randomly sampled). We notice that as the action recognition accuracy increases, the noun prediction performance also increases drastically. When the action recognition accuracy increases to 70%, we see that our method starts outperforming the AFFT baseline. However, for unseen classes, an action recognition accuracy of 55% results in performance increase. This observation also supports that accurate action recognition is needed for accurate action anticipation. Overall, we observe that with accurate action and object recognition systems, inputs in the text format can greatly improve prediction performance, without having to train modality specific encoders.

5 Conclusion and Future Work

In this work, we presented Multi-Modal Contrastive Anticipative Transformer(M-CAT), a video transformer-based approach for predictive action anticipation. We developed a two-stage process: first, contrastive pre-training between fused features from multiple modalities and rich descriptions of future actions, encoded through a text encoder; and second, fine-tuning, where the classifier (and fusion layers) are updated while predicting the future action. We evaluated and observed that object and action descriptions, added through simple text templates, can substantially improve anticipation performance. In addition, the use of richer descriptions of future actions for contrastive pre-training was beneficial. We also analyzed the effect of different modalities on performance, and the impact of the accurate actions on anticipation. In the future, we will utilize a pre-training stage similar to ImageBind [15], which learns across multiple modalities and datasets.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, 2018.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[3] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video. In *NeurIPS*, 2020.

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[6] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, pages 720–736, 2018.

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[8] Eadom Dessalene, Michael Maynord, Chinmaya Devaraj, Cornelia Fermuller, and Yiannis Aloimonos. Forecasting action through contact representations from first-person video. *TPAMI*, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[10] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *ICCV*, 2019.

[11] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *TPAMI*, 2020.

[12] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017.

[13] Rohit Girdhar and Kristen Grauman. Anticipative video transformer @ epic-kitchens action anticipation challenge 2021. In *CVPR Workshop*, 2021.

[14] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022.

[15] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[16] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[17] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, 2014.

[18] Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190, 2015.

[19] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *ICRA*, 2016.

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[21] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.

[22] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *TPAMI*, 2015.

[23] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-operative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[24] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018.

[25] Huidong Liu, Shaoyuan Xu, Jinmiao Fu, Yang Liu, Ning Xie, Chien-Chih Wang, Bryan Wang, and Yi Sun. Cma-clip: Cross-modality attention clip for image-text classification. *arXiv preprint arXiv:2112.03562*, 2021.

[26] 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.

[27] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022.

[28] M Minderer, A Gritsenko, A Stone, M Neumann, D Weissenborn, A Dosovitskiy, A Mahendran, A Arnab, M Dehghani, Z Shen, et al. Simple open-vocabulary object detection with vision transformers. arxiv 2022. *arXiv preprint arXiv:2205.06230*.

[29] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pages 529–544. Springer, 2022.

[30] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affor-dances from egocentric video. In *CVPR*, 2020.

[31] Tomislav Petković, David Puljiz, Ivan Marković, and Björn Hein. Human intention estimation based on hidden markov model motion validation for safe flexible robotized warehouses. *Robotics and Computer-Integrated Manufacturing*, 57:182–196, 2019.

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

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[34] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Pedestrian action anticipation using contextual feature fusion in stacked rnns. *arXiv preprint arXiv:2005.06582*, 2020.

[35] Debaditya Roy and Basura Fernando. Action anticipation using pairwise human-object interactions and transformers. *IEEE Transactions on Image Processing*, 30:8116–8129, 2021.

[36] Debaditya Roy and Basura Fernando. Predicting the next action by modeling the abstract goal, 2023.

[37] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020.

[38] Fadime Sener, Dibyadip Chatterjee, and Angela Yao. Technical report: Temporal aggregate representations. *arXiv preprint arXiv:2106.03152*, 2021.

[39] Sanket Thakur, Cigdem Beyan, Pietro Morerio, Vittorio Murino, and Alessio Del Bue. Enhancing next active object-based egocentric action anticipation with guided attention. *arXiv preprint arXiv:2305.12953*, 2023.

[40] Sanket Thakur, Cigdem Beyan, Pietro Morerio, Vittorio Murino, and Alessio Del Bue. Anticipating next active objects for egocentric videos. *IEEE Access*, 2024.

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[42] 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.

[43] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *TIP*, 2021.

[44] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173, 2022.

[45] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[46] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[47] Olga Zatsarynna, Yazan Abu Farha, and Juergen Gall. Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In *CVPR Workshop*, 2021.

[48] Qi Zhao, Ce Zhang, Shijie Wang, Changcheng Fu, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos? *arXiv preprint arXiv:2307.16368*, 2023.

[49] 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*, pages 6068–6077, 2023.

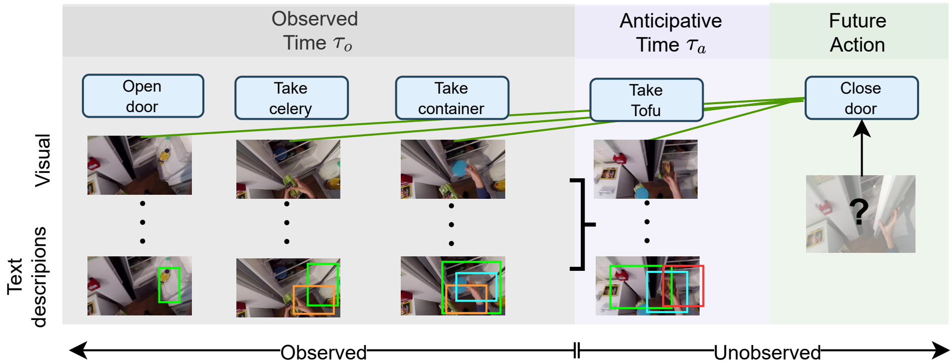

Figure 3: **Anticipating actions** τ_a seconds after observing information for τ_o seconds using multiple modalities.

A Action Anticipation

Given a video segment starting at τ_s, the objective is to predict action using a segment observed τ_o length τ_a units before it, *i.e.* from $\tau_s - (\tau_a + \tau_o)$ to $\tau_s - \tau_a$ as seen in Figure 3 (referenced in Section 3). While the anticipation time τ_a is usually fixed for each dataset, the observation time τ_o can be varied. For every i^{th} frame in the video with T frames, data from corresponding M modalities are extracted.

B Action descriptions

For action description generation, we used the ChatGPT API (GPT3.5 Turbo) to generate the descriptions. We provide examples of the descriptions generated by the ChatGPT API in Figure 4 (referenced in Sections 3.1 and 4.1). The system is asked to be an expert at video based analysis and help create a caption generation system. Then the prompt - "`Given action: {}, describe the action in 10 different ways.` is used to generate the descriptions. We see that the descriptions generated are varied, and often include other objects that are interacted with for the action to take place. For example,

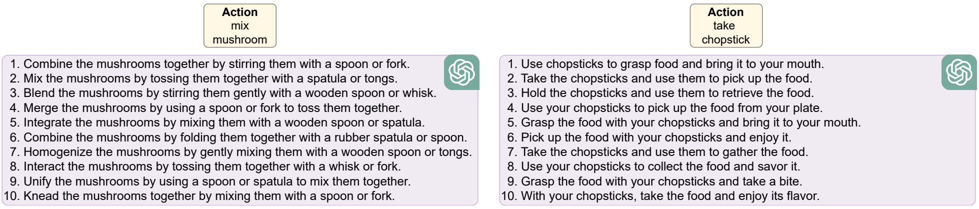

Figure 4: Descriptions generated using the ChatGPT API for actions in the EPIC-Kitchen dataset. The generated descriptions add more contextual cues for the model to learn from. For instance, for the action *take chopsticks*, the description is already alluding to the future action of “picking up food" or “eating". During training, we randomly select one description for every action.

the “Mix” action often involves the use of hands, tongs or other kitchen equipment, which are highlighted in the the descriptions. This helps our model “attend” to them in the input modalities.

C Datasets

Datasets and metrics: We evaluate our approach on three popular action anticipation datasets: *(i) Epic-Kitchens 100 (EK100)* [7], which is a large egocentric video dataset with 700 long unscripted videos of cooking activities totaling 100 hours. The dataset consists of 90K segments, and has 3807 action classes, 97 verbs and 300 nouns. We report the class-mean Recall@5 for actions, verbs and nouns; *(ii) EpicKitchens 55 (EK55)* [6] is an earlier version of Epic-Kitchens 100. For comparison to existing approaches, we report the validation accuracy on this dataset as well. EK55 has about 39K segments, and 2513 action classes, 124 verbs and 351 noun classes. For EK55, we report Top-1 and Top-5 for actions, verbs and nouns. We use the standard train and val splits to report performance. *(iii) EGTEA Gaze+* [24], an egocentric dataset containing about 10K segments, and 19 verbs, 51 nouns and 106 unique actions. Following [13], we report the performance on the first split of the dataset at $\tau_a = 0.5$s. We report the Top-1 and class-mean(cm) Top-1 accuracies for actions, nouns and verb.

D Baselines

In addition to comparing our method to its variants containing different modalities, we also evaluate against the state-of-the-art for action anticipation, including: RULSTM [11], AVT [13], ActionBanks [37], AFFT [49], and MeMViT [42]. RULSTM [11] leverages a ‘rolling’ LSTM to encoder the past and an ‘unrolling’ LSTM to predict the future. ActionBanks [37] improves over RULSTM by carefully leveraging long-term action blocks and non-local blocks. AVT [13] uses an attention-based video modelling architecture that attends to previous frames to anticipate the future. MeMViT [42], on the other hand, processes videos online by using cache “memory", through which the model learns to refer prior context for long-term anticipation. AFFT [49] improves on AVT by using multiple modalities, and using self-attention modules to fuse the features together.

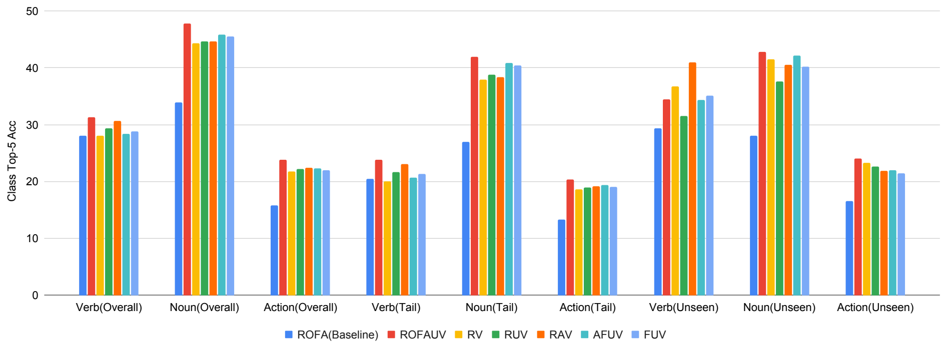

Figure 5: Recall@5 for verb, noun and actions on the EK100 dataset for different modality combinations. The first bar is the baseline (i.e., AFFT) using (ROFA) modalities. Objects and actions are used as input by converting them to text through – "A video containing the objects/actions <xyz>", and embeddings from the text encoder are used in the fusion module. ROFAUV stands for - R(RGB), O(Obj features), F(Flow), A(Audio), U(FasterRCNN detected objects in text form), V(Actions in text form) modalities

E Contribution of Each Modality on Action Anticipation

In Figure 5, we explore the contributions of various modalities to performance. For all the experiments, we use the objects provided by [ ], and ground truth labels for actions.

We first compare our model performance ROFAUV against ROFA (also noted in Table 3 of the main paper). We see that the additional modalities *i.e* Objects and Actions significantly improve the performance.

With RGB (R) as a base modality, comparing RV with RUV, we see that the objects detected using fasterRCNN model aid in the performance, however, through a small margin. To understand the impact of using object information as an additional modality, we examine the detected objects and the actions in Table 5. We see that for rows 1 and 3, the object required for the action prediction is not detected by the FasterRCNN model with high probability. For rows 2 and 4, while the object was detected, presence of other objects make the action prediction challenging. On the other hand, actions (which are often defined as a verb-noun pair) give more information about the objects being interacted and the actions in the observed frames. Therefore, while detecting objects accurately is essential and makes one part of the action (<verb,noun>), it is also vital that an active hand-object interaction be detected.

Comparing RUV, FUV and AFUV we see that audio and flow also aids in the model performance, and in combination provide the similar information to the model as the RGB data.

F Using GPT-4 to refine predictions

For EK55, we also explore using ChatGPT (GPT-4) to reason about the future action, given a sample set of examples from the train set, and a list of actions to choose from. We provide the Top-10 actions predicted by our model, and ask ChatGPT to pick the most likely action,

	FasterRCNN objects	Actions	Future Action
1	'sponge, tap', 'sponge, tap','sponge, tap','sponge, tap', 'sponge, tap','sponge, tap','sponge, tap', 'sponge, tap','sponge, tap','sponge, tap'	'wash plate', 'wash plate', 'no action', 'wash plate', 'wash plate', 'wash plate', 'wash plate', 'wash plate','insert plate', 'insert plate', 'wash sponge','wash sponge','wash sponge', 'wash sponge','wash sponge', 'wash sponge'	Wash cloth
2	'bin, spoon', 'bin', 'knife, ', 'knife, ', 'bin', 'bin', 'bag', 'bag', 'bag', 'bin, bag', 'bin, bag','bin, bag', 'bin, bag', 'bin, bag', 'bin, bag', 'bin, bag'	'wrap bag', 'wrap bag', 'wrap bag', 'wrap bag','wrap bag', 'wrap bag', 'wrap bag', 'wrap bag','wrap bag', 'wrap bag', 'wrap bag', 'wrap bag','wrap bag', 'wrap bag', 'wrap bag'	Tie Bag
3	'cupboard', 'cupboard', 'cupboard', 'cupboard', 'cupboard', 'pan,cupboard','pan,cupboard', 'cupboard', 'cupboard', 'cupboard, lid', 'pan,cupboard', 'pan,cupboard','pan,cupboard', 'pan,cupboard', 'pan,cupboard', 'pan, '	'take plate', 'take plate', 'take plate', 'take plate', 'take plate', 'no action', 'open cupboard', 'no action', 'insert plate', 'insert plate', 'no action', 'no action', 'take cup','no action', 'open cupboard', 'insert cup'	Put-into Cup
4	'bowl,spoon, tap, knife', 'bowl,spoon, tap, knife', 'bowl,spoon, tap, knife','bowl, spoon, cup, tap, knife', 'bowl, spoon, tap, knife', 'bowl, spoon, tap, knife','bowl, spoon, cup, knife, bottle', 'bowl, cup, tap, knife, lid', 'bowl,knife, tap','bowl, spoon, tap, knife, lid', 'bowl, spoon, tap, knife', 'bowl,knife, tap','bowl, spoon, tap, knife', 'bowl, spoon, tap, knife, sponge', 'knife,tap, spoon'	'wash cup', 'wash cup', 'no action', 'wash spoon', 'wash spoon', 'put spoon', 'wash cup', 'wash cup', 'wash cup','wash cup', 'wash cup', 'wash cup', 'wash cup', 'no action', 'no action', ' turn-off tap'	Turn-off tap
5	'board:chopping, onion, knife, spatula', 'board:chopping, onion, knife, spatula', 'knife, onion, spatula', 'knife, board:chopping, onion, spatula', 'food, onion, knife, spatula','knife, board:chopping, onion, spatula', 'knife, board:chopping, onion, spatula','knife, board:chopping,spatula', 'knife, board:chopping, lid, spatula', 'board:chopping,knife', 'knife, board:chopping,spatula',	'put board:chopping', 'no action', 'no action', 'put knife', 'no action', 'open drawer', 'no action', 'no action','take spatula', 'close drawer', 'no action', 'mix aubergine', 'mix aubergine', 'mix aubergine', 'mix aubergine', 'put aubergine'	Take salt
6	'lid, glass, bottle', 'lid', 'lid, glass', 'glass, lid', 'glass, lid', 'glass, container', 'glass, bottle, container', 'bag, glass', 'bag, glass', 'glass, bag, bottle', 'glass, bottle, oil',' glass, bag, bottle, container', 'glass, bag, lid, bottle', spoon, bottle, glass, bag, lid', 'onion, lid', 'cup, glass, bottle, bag, lid'	'move bin', ' move bin', 'take milk', 'take milk',' crush milk', 'crush milk', 'crush milk', 'crush milk','no action', 'no action', 'no action', ' insert milk','no action', 'no action', 'take paper', 'take paper'	Move glass

Table 5: Per frame objects and actions detected in a video clip in EPICKitchens-100 dataset. The objects are detected using FasterRCNN trained on EK55 dataset. We set a threshold of 0.15, and select only top 5 objects per frame. Actions described here are the ground truth annotations. When actions are not detected, a 'no action" label is used instead.

given the actions in the observed timeframe (see Figure 6 for reference). The previous observed actions are the ground truth, and the options are the top-10 results generated by the model using the ROFA modalities. We provide examples in Table 7.

From Table 6, we observe that while the Top-5 and recall performance improve for Verb, Noun and Actions, the Top-1 performance drops for all predictions. As we see in Table 7, while GPT-4 can refine the predictions in some cases, it also produces incorrect answers especially in cases where several "no action" are present (as the predictions are solely based on the text input of the past actions). When the correct action is not present in the Top-5, but is in Top-10, GPT-4 is capable of refining the prediction. It should be noted when the correct action is not present in the Top-10 predictions at all, GPT-4 is not capable of predicting the right action, as we force it to pick one of the 10 actions.

```
You are a helpful AI assistant to predict the next most probable next action based on the observed actions and
common sense. The given previous observed actions are in the form of a sequence of action pairs, each action
pair is defined by a {verb} and a {noun}, separated by a space.


E1: 'mix vegetable', ... 'put spatula', 'no action', 'take tofu', 'no action' => "take knife"
E2: 'take pan', ... 'filter pan', 'filter rice', 'filter rice', 'filter rice' => "pour rice"
E3: 'wash pan', ... 'put pan', 'wash cloth', 'wash sink', 'wash sink' => "wash cloth"
E4: 'put mushroom', 'move liquid:washing',... , 'wash mushroom', 'wash mushroom' => "take mushroom"
Q: 'take_salt', 'no action', ... 'put-down_salt', 'no action' =>

Options: ['open_door', 'take_fork', 'take_plate', 'take_salt', 'take_spoon', 'put-down_fork', 'open_drawer',
'rinse_hand', 'take_spatula', 'take_napkin']


Please predict the future action only from the options presented.
The output should be in the json format, with the predicted action as the key.
```

Figure 6: Prompt provided to ChatGPT to pick most plausible future action.

Method	Verb		Noun		Action		
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Recall@5
Ours (baseline)	**32.8**	79.8	**27.8**	56.5	**15.6**	36.8	16.1
Ours (GPT corrected)	28.4	**79.9**	25.2	**57.1**	13	**37.6**	**16.5**

Table 6: EK55: Accuracy after using GPT-4 to correct the predictions.

Observed Actions	Future Action (corrected)	Future Actions (predicted)	GT
no action, no action, open_door, no action, take_container, take_container, no action, take_lid, take_lid, no action, no action, no action, no action, take_lid, take_lid, take_lid	put lid	close_door, take_pan, put lid, put-down_pan, open_door, put-down_box:cereal, take_box:fruit, take_colander, put-down_colander, take_bag:cereal	put lid
close_fridge, close_fridge, no action, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal, open_bag:cereal	take_bowl	fold_bag:rice, put-down_bag, close_bag:rice, open_bag:cereal, take_bag:cereal, place_salad, put_packet:crisp, get_salad, take_bowl, put-in_bag	open_bag:cereal
stir_spatula, stir_spatula, put-down_spatula, open_container, open_container, take_onion, take_onion, take_onion, take_onion, close_container, close_container, close_container, no action, no action, take_spatula, take_knife	cut_onion	put-down_spatula, take_spatula, open_container, put-down_onion, put-down_knife, put_container, cut_onion, take_container, close_container, take_onion	cut_onion
take_salt, no action, no action, no action, put-down_salt, put-down_salt, put-down_salt, put-down_salt, put-down_salt, put-down_salt, no action, no action, no action, no action, no action, no action	take_spoon	open_door, take_fork, take_plate, take_salt, take_spoon, put-down_fork, open_drawer, rinse_hand, take_spatula, take_napkin	open_door

Table 7: Examples showing the past observed actions, GPT4 corrected action, predicted actions by our model, and ground-truth. Actions in the “Future Actions (predicted)” are in descending order of probability.