Open-Vocabulary Long-term Action Anticipation

Syed Talal Wasim*[1,2], Jinhui Yi*[1,2], Hamid Suleman[1,2], Ahmad Javed[1], Yanan Luo[1], Muzammal Naseer[3,4], and Juergen Gall[1,2]

[1] University of Bonn, Germany
[2] Lamarr Institute for Machine Learning and Artificial Intelligence, Germany
[3] Khalifa University, Abu Dhabi, UAE
[4] University of Western Australia, Australia
* Equal Contribution

Abstract. Action anticipation, i.e., predicting future actions from past video observations, is fundamental to intelligent systems that assist humans. Despite progress in model architectures, current evaluation practices exhibit critical limitations: methods evaluate exclusively in closed-set settings where the training and test action vocabularies are identical, thereby preventing an understanding of generalization to novel action classes encountered in real-world deployment. We therefore introduce the first open-vocabulary evaluation framework for action anticipation, where models are trained on one egocentric dataset and tested on entirely different egocentric datasets with novel action vocabularies. Since our thorough evaluation shows that adapting existing approaches to this task is insufficient, we propose a novel approach that employs horizon-specific learnable queries and a lightweight text encoder adaptation for open-vocabulary long-term action anticipation. It substantially outperforms other approaches that we have adapted to this task.

1 Introduction

Action anticipation, the task of predicting future actions from past video observations, is fundamental to building intelligent systems that can proactively assist humans in daily activities, enable safer human-robot collaboration, and power next-generation assistive technologies. For egocentric videos and long time horizons, e.g., predicting the next 20 future actions as in the long-term action anticipation challenge of the egocentric Ego4D dataset [14], this task is very challenging. Despite significant progress in model architectures for action anticipation [21, 50], the methods so far only forecast actions that have been part of the training data. This closed-set setting, where actions are defined by a limited set of pairs of verbs and nouns observed during training, limits their ability to generalize to novel action classes, a crucial capability for real-world deployment where the space of possible actions is inherently open-ended and cannot be exhaustively enumerated during training.

To address this limitation, we introduce the first open-vocabulary evaluation protocol for long-term action anticipation as illustrated in Fig. 1. Our protocol enables rigorous assessment of model generalization both within the training

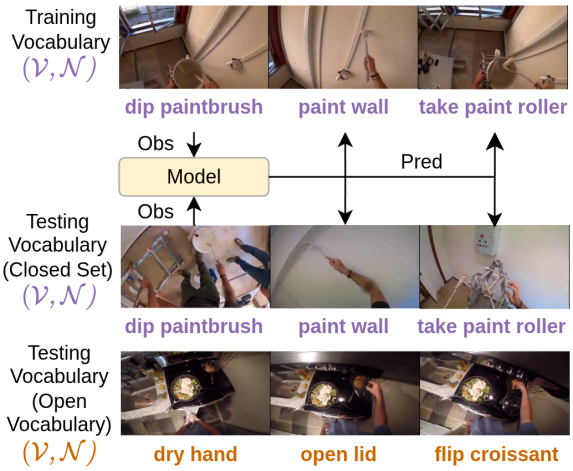

Fig. 1: Open-vocabulary action anticipation: In the commonly used closed-set protocol, the actions, which are represented by a vocabulary of verbs $\mathcal{V}$ and nouns $\mathcal{N}$, are shared across the training and test set. In this work, we introduce the first open-vocabulary action anticipation protocol where the overlap between the actions in the training and test set is small, as it is the case for many practical scenarios.

distribution (in-domain evaluation) and to unseen action classes from different datasets (out-of-distribution evaluation). To this end, we follow the long-term anticipation protocol of Ego4D [14] and use this dataset for training. In contrast to closed-set action anticipation that uses Ego4D for evaluation as well, we use EPIC-Kitchens [6] and EGTEA+ [22] for our open-vocabulary evaluation protocol as test datasets. As shown in Fig. 2, there is very little overlap of the actions between the training and test set under this protocol.

Since open-vocabulary long-term action anticipation has not been addressed before, we adapt various models to the new task, namely AntGPT [47], Video-LLaVA [23], XCLIP [30], and LaViLa [48]. While we fine-tune AntGPT and Video-LLaVA on the training set, we augment XCLIP and LaViLa using separate prediction heads for each future action as in [14,25]. However, we show that these methods do not perform very well. Either they tend to overfit to the actions of the training set, or the multiple heads do not generate a coherent sequence of future actions. To address the latter, we furthermore propose a novel architecture tailored for open-vocabulary long-term action anticipation. Instead of using multiple separate heads to generate a sequence of future actions, we propose a single Multi-Horizon Cross-Attention Decoder (MHCAD) that utilizes distinct learnable queries for each future action. Combined with a lightweight adaptation of dual text encoders for verbs and nouns, our model can be applied to both closed-set and open-vocabulary settings. In summary, our main contributions are:

- We introduce the first open-vocabulary evaluation framework for action anticipation, enabling systematic assessment of model generalization to both in-distribution and out-of-distribution action classes across multiple datasets.
- We propose MHCAD, a novel architecture with horizon-specific learnable queries. For closed-set action anticipation, it outperforms methods within the same parameter range, and it is on-par with much larger models that have 12 times more parameters. For open-vocabulary action anticipation,

it substantially outperforms other approaches that we have adapted to this task.

2 Related Work

2.1 Action Anticipation

Protocols: Action anticipation has been formulated through several distinct evaluation protocols, each emphasizing different aspects of temporal reasoning. Next-action anticipation [9, 11] focuses on predicting the immediate next action following an observed video clip, testing a model's ability to capture local temporal patterns. Last-action anticipation [28, 29] requires predicting the final action that will occur significantly later in the video, evaluating whether models can reason about high-level goals and long-range dependencies. The Ego4D benchmark [14] introduced long-term anticipation, which generalizes both paradigms by requiring models to predict Z future actions in chronological order at multiple time horizons (e.g., 20 future timestamps). This protocol subsumes both next-action and last-action anticipation as special cases while additionally challenging models to maintain correct sequential ordering. Despite this comprehensive formulation, existing work exhibits fragmented evaluation practices: methods report separately on next-action and last-action tasks [26, 47] despite both being subsets of the long-term anticipation protocol, and evaluate on inconsistent dataset versions (e.g., Ego4D v1 validation vs. v2 test), hindering reliable cross-model comparison.

Next-Action Anticipation: Next-action anticipation methods address the challenge of predicting the single immediate action (typically a verb-noun pair) following an observed clip. RULSTM [9] introduced recurrent mechanisms with residual connections to model temporal dynamics, while [11] proposed an end-to-end attention model that jointly predicts next actions and future frame features. UADT [15] achieves strong performance through separate encoder-decoder architectures for verbs and nouns, dynamically combining both streams to capture compositional action structure. Recent work has explored specialized mechanisms: InAViT [33] models human-object interactions via spatial and trajectory cross-attention, AFFT [51] employs attention-based feature fusion, and MemViT [42] introduces memory-augmented multiscale vision transformers. Another recent work, CLAM [49], proposes a linear-time, constant-memory module that retrieves complementary context cues from frame features using cross-attention over a linear attentive-memory for scalable long-video action anticipation. Temporal aggregation strategies have been investigated by [10, 34], with the latter presenting a two-stage architecture using shuffled causal masking for latency-aware forecasting. Foundation model approaches, such as [5,26] have also reported results on this task.

Last-Action Anticipation: Last-action anticipation requires predicting the final action in a long-term activity sequence given an early observation. Early methods adapted from action recognition [4, 12, 16, 17] established baselines for this task. Ego-Topo [28] leveraged topological priors from egocentric interaction

patterns, while Anticipatr [29] combined segment-level representations learned across activities with video-level representations. Among foundation model approaches, AntGPT [47] has reported results on last-action anticipation alongside other anticipation tasks.

Long-Term Anticipation: The Ego4D benchmark [14] introduced long-term anticipation, which requires predicting sequences of future actions at multiple time horizons. Early approaches established baselines using conventional architectures: ICVAE [25], HierVL [3], VCLIP [7], SlowFast [14], and the CLAM baseline [49], demonstrating the feasibility of long-term anticipation through carefully designed video encoders and temporal modeling. Recent work has increasingly leveraged large-scale foundation models: PlausiVL [26] employs LLaMA-2-7B [36] with Q-former and ViT-g for temporal modeling, AntGPT [47] uses LLaMA-2-13B/7B for goal inference and temporal dynamics, PaLM [19] combines action recognition with context-driven prompting using LLaMA-2-7B, and VideoLLM [5] leverages modality encoding with large language models. However, these foundation model approaches operate at different scales than conventional vision models, often incorporating multimodal aggregators whose parameter counts exceed entire task-specific networks. Their performance reflects extensive scaling and pretraining rather than task-specific architectural innovation.

Dense Anticipation: Dense anticipation predicts the labels and temporal extents of all future actions over a portion of the unobserved video, effectively performing temporal action segmentation of the future: rather than forecasting only the next action, the model labels every upcoming frame. This requires densely annotated benchmarks in which a single action occurs at any given time. Approaches are typically categorized by their output: deterministic methods [2, 13, 18, 34] commit to a single future trajectory, whereas stochastic methods [1, 41, 46] model the ambiguity of the future by sampling multiple continuations, using recurrent, adversarial, or diffusion-based generators. A separate family [44, 52] supports both inference modes while unifying past action recognition with future forecasting. Since dense anticipation is evaluated under different protocols and benchmarks than those used in this work, we do not compare directly against these methods.

2.2 Vision-Language Models for Videos

The foundational work of CLIP [32] demonstrated the efficacy of large-scale vision-language pretraining on image-text pairs, establishing a powerful paradigm for zero-shot generalization. Early extensions of CLIP to video understanding focused on adapting its architecture to capture temporal dynamics. Models such as ActionCLIP [37], VitaCLIP [40], and XCLIP [30] modified the image encoder to enable action recognition in videos. More recent generalist models such as InternVideo [38, 39] leverage massive video-text datasets to train sophisticated spatio-temporal architectures, achieving state-of-the-art results across diverse video understanding tasks.

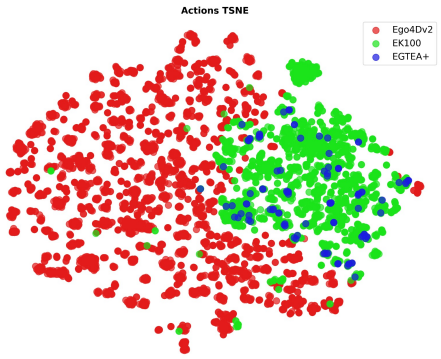

Fig. 2: t-SNE visualization of text-encoded unique actions (verb-noun pairs): We encode actions from the Ego4D v2 training set (red), EK100 validation set (green), and EGTEA+ validation set (blue) using our adapted text encoder. There are only very few samples where the actions overlap between the training (red) and test datasets (green or blue).

Specialized vision-language models have been developed specifically for egocentric videos. Egocentric Video-Language Pretraining [24] introduced the EgoNCE contrastive learning objective that mines egocentric-aware samples from large-scale first-person datasets derived from Ego4D. EgoVLPv2 [31] extended this approach by incorporating cross-modal fusion directly into video and language backbones to learn stronger video-text representations. LaViLa [48] leveraged large language models as automatic video narrators to generate dense, diverse narrations for contrastive video-language representation learning. These egocentric-specific models capture the unique characteristics of first-person visual perspective and interaction patterns.

3 Open-Vocabulary Long-term Action Anticipation

We address the long-term action anticipation task, which has been introduced in [14]. Given the last O observed actions $\{\mathbf{a}^1, \ldots, \mathbf{a}^O\}$ and the corresponding video segments $\{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^O\}$, the task is to predict the next Z future actions $\{\hat{\mathbf{a}}^{O+1}, \ldots, \hat{\mathbf{a}}^{O+Z}\}$ where each action is represented by a verb and noun pair. So far, this task has been evaluated only on the Ego4D dataset [14] and in a closed-set setting, i.e., all action labels are part of the training data. In this work, we extend this protocol to an open-vocabulary setting where we use Ego4D as training data and evaluate the trained model on other egocentric datasets. This means that it requires forecasting also actions that are not part of Ego4D.

Datasets: We evaluate our method on three egocentric action anticipation datasets. **Ego4D** [14] is a large-scale egocentric dataset covering diverse indoor and outdoor scenarios, including homes, workplaces, and public spaces. It consists of 3,670 hours of videos with 117 verbs and 521 nouns. We use videos from the Forecasting and Hand-Object interaction subset. **EPIC-Kitchens-100** [6] is an egocentric dataset captured in kitchen environments, consisting of 100 hours of videos with 97 verbs and 300 nouns. **EGTEA+** [22] contains approximately 26 hours of egocentric cooking videos with 19 verbs and 53 nouns. For EPIC-Kitchens-100 and EGTEA+, we employ the same train and test splits as described in [28].

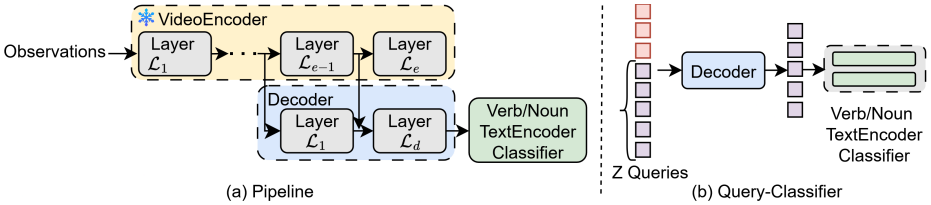

Fig. 3: Model overview: (a) Our pipeline extracts multi-scale features from the last L_d layers of a frozen vision encoder and processes them through a Multi-Horizon Cross-Attention Decoder with L_d layers. Predictions are made via open-vocabulary matching against dual text encoders. (b) The decoder uses Z horizon-specific queries (purple) together with O CLS tokens (red). The shared verb/noun classification heads match visual embeddings for each horizon-specific query to text-encoded action classes.

Closed-Set Evaluation: In the closed-set setting, we train and evaluate on Ego4D using identical action vocabularies for training and testing. We report results on three evaluation sets: Ego4D v1 validation, v1 test, and v2 test. These represent the different sets used by various prior works, and we evaluate on all three to enable a comprehensive comparison with existing methods. Following the standard Ego4D long-term anticipation protocol, we observe $O = 8$ video clips and predict $Z = 20$ future actions. We report edit distance as an evaluation metric, where lower values indicate better performance.

Open-Vocabulary Evaluation: The open-vocabulary evaluation protocol assesses the generalization to novel action classes. We train exclusively on Ego4D v2 and evaluate on the EPIC-Kitchens-100 and EGTEA+ validation sets without any training on their respective training data. As visualized in Fig. 2, there is very little overlap between the actions in Ego4D v2 and the actions in EPIC-Kitchens-100 or EGTEA+. This protocol tests whether models can generalize to entirely unseen action vocabularies from different domains and datasets. To unify evaluation across datasets, we adopt the same temporal structure as Ego4D: we observe $O = 8$ clips and predict $Z = 20$ future actions for both EPIC-Kitchens-100 and EGTEA+, and report edit distance as the evaluation metric. This contrasts with traditional evaluation protocols on these datasets, which focus on single-step next-action or last-action prediction.

4 Method

To address open-vocabulary long-term action anticipation for the first time, we adapt various methods to the new task. For instance, we combine the video encoder LaViLa [48] with a multi-head decoder as it is commonly used for long-term action anticipation in closed settings [7, 14, 25]. The decoder takes the aggregated video representation and feeds it into $2 \times Z$ prediction heads. Each head then predicts the verb or noun for each future action $\hat{\mathbf{a}}^i$. This approach, however, suffers from a fundamental limitation. It relies on a single aggregated representation to make predictions across all Z future horizons by separate pre-

diction heads, which limits the ability of making consistent predictions across all Z future actions.

We thus propose an alternative decoder as illustrated in Fig. 3. It uses horizon-specific queries and outputs embedding vectors for each future verb-noun pair, which are then matched against text encoder features of verbs and nouns via dot products. In this way, the decoder can be applied to both closed-set and open-vocabulary long-term action anticipation by generating verb and noun pairs that have not been present in the training set. While we describe the decoder in Section 4.1, the text encoder with a lightweight adaptation mechanism is described in Section 4.2.

4.1 Multi-Horizon Cross-Attention Decoder

As illustrated in Fig. 4, our approach extracts multi-layer dense features from observed segments and employs a cascade of cross-attention layers to progressively aggregate information. Critically, our decoder maintains separate query tokens for each anticipation horizon, enabling horizon-specific feature aggregation rather than forcing all predictions to stem from a single representation. We thus term our decoder *Multi-Horizon Cross-Attention Decoder* (MHCAD). We describe each step in detail.

Visual Feature Extraction: Given O observed video segments $\{\mathbf{S}^1, \ldots, \mathbf{S}^O\}$, we first extract features using a frozen pretrained vision transformer backbone [48]. For each video segment $\mathbf{S}^i$, we uniformly sample T frames and pass them through the backbone, which produces a CLS token $\mathbf{c}^i \in \mathbb{R}^d$ and spatial feature maps from the last L_d layers.

For each layer ℓ and segment $\mathbf{S}^i$, we obtain spatial features $\mathbf{F}_\ell^i \in \mathbb{R}^{T\times P\times d}$, where P denotes the number of spatial tokens per frame and d is the feature dimension. We exclude the CLS tokens from these features and concatenate across all observed segments:

$$\mathbf{F}_\ell = [\mathbf{F}_\ell^1; \ldots; \mathbf{F}_\ell^O] \in \mathbb{R}^{(O\cdot T\cdot P)\times d}. \tag{1}$$

The tokens $\mathbf{F}_\ell$ provide additional fine-grained information compared to the CLS tokens, which is in particular relevant for open-vocabulary action anticipation, as we demonstrate in the experimental evaluation.

To enable the model to distinguish between tokens based on their hierarchical spatio-temporal structure—specifically, their position in the segment sequence (i), their temporal order within segments (t), and their spatial location within frames (h, w)—we apply 4D Rotary Position Embeddings (RoPE) [35]. Unlike standard learned positional embeddings, RoPE encodes relative positions through rotation operations in the complex plane, enabling better length generalization while maintaining computational efficiency through a factorized representation [35]. For each spatial token in $\mathbf{F}_\ell$, we compute its position tuple (i, t, h, w) where $i \in \{1, \ldots, O\}$ is the segment index, $t \in \{1, \ldots, T\}$ is the frame index, and (h, w) are the 2D spatial coordinates. We partition the d-dimensional feature vector into four equal parts of dimension $d' = d/4$ each, and apply independent rotary embeddings to each part:

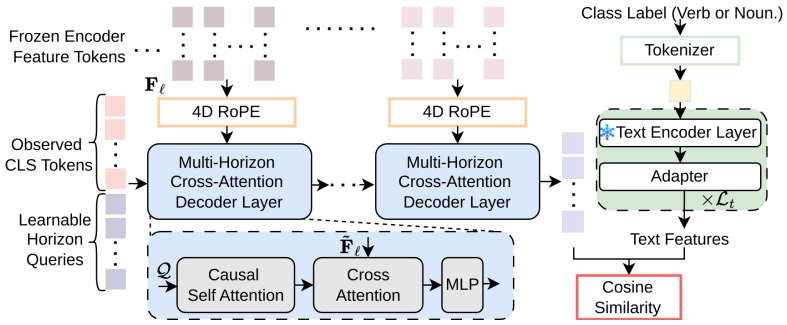

Fig. 4: Multi-Horizon Cross-Attention Decoder architecture: Multi-scale spatial features $\mathbf{F}_\ell$ from encoder layers are encoded with 4D RoPE and fed to L_d decoder layers. Each layer performs causal self-attention on queries (observed CLS tokens and horizon-specific learnable queries) followed by cross-attention with encoder features and an MLP. Final horizon queries are projected and matched against text features from adapted frozen text encoders via cosine similarity for open-vocabulary prediction.

$$\tilde{\mathbf{F}}_{\ell,j} = [\mathrm{RoPE}_i(f^1_{\ell,j}); \mathrm{RoPE}_t(f^2_{\ell,j}); \tag{2}$$

$$\mathrm{RoPE}_h(f^3_{\ell,j}); \mathrm{RoPE}_w(f^4_{\ell,j})] \tag{3}$$

where $f^k_{\ell,j} = \mathbf{F}^{(k-1)d'+1:kd'}_{\ell,j}$, i.e., the k-th quarter of the feature vector for token j. This yields position-encoded feature maps $\tilde{\mathbf{F}}_\ell \in \mathbb{R}^{(O\cdot T\cdot P)\times d}$ for each encoder layer ℓ.

Query Initialization: As illustrated in Fig. 4, our decoder maintains two types of query tokens: (1) O segment-level queries initialized from the observed CLS tokens $\{\mathbf{c}^1, \ldots, \mathbf{c}^O\}$, and (2) Z horizon-specific learnable queries $\{\mathbf{q}^1, \ldots, \mathbf{q}^Z\}$ corresponding to each of the future actions $\{\hat{\mathbf{a}}^{O+1}, \ldots, \hat{\mathbf{a}}^{O+Z}\}$. These are concatenated to form the initial query sequence:

$$\mathbf{Q}^{(0)} = [\mathbf{c}^1; \ldots; \mathbf{c}^O; \mathbf{q}^1; \ldots; \mathbf{q}^Z] \in \mathbb{R}^{(O+Z)\times d}. \tag{4}$$

While the segment-level queries provide contextualized representations of observed segments, the learnable queries are optimized for their respective future horizons. This is in contrast to previous works that do not consider the sequence of future actions through the entire decoder but only at the end of the decoder by using for each future action a separate head. Since future actions that occur later in the sequence might need to attend to different observed features than actions that occur earlier, it is important to consider the different future horizons already from the beginning of the decoder, as we show in the experiments.

Multi-Layer Cross-Attention Aggregation: Starting with the initial queries $\mathbf{Q}^{(0)}$, our decoder, consisting of L_d layers, progressively refines the query representations through alternating self-attention and cross-attention operations. At each decoder layer m, we first apply causal self-attention to enable information

flow among query tokens while preserving temporal causality:

$$\mathbf{Q}_{\text{self}}^{(m-1)} = \text{CausalSelfAttention}(\mathbf{Q}^{(m-1)}) \tag{5}$$

where the causal mask ensures that each query token at position j can only attend to positions $\leq j$, preventing future queries from accessing information about later horizons.

We then perform cross-attention between the self-attended queries and visual features from the corresponding encoder layer. Specifically, at decoder layer m, we use features from encoder layer $\ell = L_e - L_d + m$, where L_e is the number of layers of the encoder, forming the key-value pairs $\mathbf{K}_\ell = \mathbf{V}_\ell = \tilde{\mathbf{F}}_\ell \in \mathbb{R}^{(O \cdot T \cdot P) \times d}$.

The cross-attention is then computed between the previous queries and the corresponding position-encoded tokens:

$$\mathbf{Q}'^{(m)} = \text{CrossAttention}(\mathbf{Q}_{\text{self}}^{(m-1)}, \mathbf{K}_\ell, \mathbf{V}_\ell). \tag{6}$$

Finally, we update the queries through a residual connection with stochastic depth and a feed-forward network:

$$\mathbf{Q}_{\text{res}}^{(m)} = \mathbf{Q}^{(m-1)} + \text{DropPath}(\mathbf{Q}'^{(m)}) \tag{7}$$

$$\mathbf{Q}^{(m)} = \mathbf{Q}_{\text{res}}^{(m)} + \text{FFN}(\text{LayerNorm}(\mathbf{Q}_{\text{res}}^{(m)})) \tag{8}$$

where DropPath randomly drops the residual branch during training with probability p_{drop}.

This multi-layer design enables progressive refinement of query representations, with each decoder layer attending to features at different semantic levels from the visual encoder and enabling each horizon-specific query to attend to different tokens.

Prediction Heads: After L_d decoder layers, we extract the horizon-specific queries $\{\mathbf{q}^1, \ldots, \mathbf{q}^Z\}$ from the final output $\mathbf{Q}^{(L_d)}$ and apply two shared projection heads, one each for verbs and nouns, to generate embeddings for each horizon:

$$\mathbf{v}_v^j = h_v(\mathbf{q}^j) \in \mathbb{R}^{d_e}, \quad \mathbf{v}_n^j = h_n(\mathbf{q}^j) \in \mathbb{R}^{d_e}, \tag{9}$$

where h_v and h_n are linear projection heads for verbs and nouns respectively, d_e is the embedding dimension of the text encoder, and $j \in \{1, \ldots, Z\}$. These shared projection heads enable the model to learn unified verb and noun representations across all future time steps. The resulting embeddings are then matched against text features as described in Section 4.2.

4.2 Text Encoder with Lightweight Adaptation

To enable open-vocabulary anticipation, we employ dual text encoders, one for verbs and one for nouns, based on the BERT-style [8] transformer architecture commonly used in CLIP models [32]. Both encoders have L_t transformer layers

and remain frozen during training, with adaptation achieved through bottleneck adapters.

Bottleneck Adapters: We enhance the text encoders' adaptability by inserting lightweight bottleneck adapter modules [43] after each of the L_t transformer blocks. An adapter consists of a projection to a lower-dimensional bottleneck, followed by a nonlinearity and a projection back to the original dimension:

$$\mathbf{h}_{\text{down}} = \text{GELU}(W_{\text{down}}\mathbf{h}) \tag{10}$$

$$\mathbf{h}_{\text{adapt}} = W_{\text{up}}\mathbf{h}_{\text{down}} \tag{11}$$

$$\mathbf{h}_{\text{out}} = \mathbf{h} + \alpha \cdot \mathbf{h}_{\text{adapt}} \tag{12}$$

where $W_{\text{down}} \in \mathbb{R}^{d_{\text{bottleneck}} \times d}$ and $W_{\text{up}} \in \mathbb{R}^{d \times d_{\text{bottleneck}}}$ are learnable projection matrices with $d_{\text{bottleneck}} \ll d$, and α is a learnable scalar that controls the adapter's contribution. This design enables task-specific adaptation while keeping the vast majority of the pretrained parameters frozen [43].

Text Feature Extraction and Matching: For each class (verb or noun), we encode its text representation through the adapted encoder and extract the final CLS token as the class embedding. Let $\mathcal{V} = \{v_1, \ldots, v_{N_v}\}$ and $\mathcal{N} = \{n_1, \ldots, n_{N_n}\}$ denote the sets of verb and noun classes, respectively. We then obtain text embeddings:

$$\mathbf{t}_v^k = \text{TextEncoder}_v(v_k) \in \mathbb{R}^{d_e}, \quad k \in \{1, \ldots, N_v\}, \tag{13}$$

$$\mathbf{t}_n^k = \text{TextEncoder}_n(n_k) \in \mathbb{R}^{d_e}, \quad k \in \{1, \ldots, N_n\}. \tag{14}$$

Following the CLIP paradigm [32], we compute prediction logits through normalized dot products between visual and text embeddings:

$$\mathbf{s}_v^j = \tau \cdot \frac{\mathbf{v}_v^j}{\|\mathbf{v}_v^j\|} \cdot \left[\frac{\mathbf{t}_v^1}{\|\mathbf{t}_v^1\|}, \ldots, \frac{\mathbf{t}_v^{N_v}}{\|\mathbf{t}_v^{N_v}\|}\right]^\top \in \mathbb{R}^{N_v} \tag{15}$$

$$\mathbf{s}_n^j = \tau \cdot \frac{\mathbf{v}_n^j}{\|\mathbf{v}_n^j\|} \cdot \left[\frac{\mathbf{t}_n^1}{\|\mathbf{t}_n^1\|}, \ldots, \frac{\mathbf{t}_n^{N_n}}{\|\mathbf{t}_n^{N_n}\|}\right]^\top \in \mathbb{R}^{N_n} \tag{16}$$

where τ is a learnable temperature parameter. This formulation enables zero-shot generalization to unseen action classes at test time.

4.3 Loss Function and Training

We train our model using the standard cross-entropy loss for both verb and noun predictions at each anticipation horizon. Let y_v^j and y_n^j denote the ground-truth verb and noun labels for the j-th future segment. The loss is then given by

$$\mathcal{L} = \frac{1}{Z}\sum_{j=1}^{Z}\left[\mathcal{L}_{\text{CE}}(\mathbf{s}_v^j, y_v^j) + \mathcal{L}_{\text{CE}}(\mathbf{s}_n^j, y_n^j)\right] \tag{17}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss, and the average over horizons ensures balanced optimization across near and distant futures.

Table 1: Comparison with state-of-the-art methods on Ego4D for the Closed-Set Evaluation. The LLM/VLM models are greyed out to highlight that these methods are not directly comparable since they use 7B or more parameters. * indicates parameter counts estimated from paper details and/or code when not explicitly provided by authors.

Method	Frozen Params	Trained Params	Ego4D v1 Val		Ego4D v1 Test		Ego4D v2 Test	
			Verb ↓	Noun ↓	Verb ↓	Noun ↓	Verb ↓	Noun ↓
LLM/VLM Foundation Models with >7B parameters								
VideoLLM [5]	7.0B*	67M*	-	-	-	-	0.721	0.725
VideoLLaMA [45]	8.2B*	42M*	0.703	0.721	-	-	-	-
PlausiVL [26]	7.74B*	188M*	0.679	0.681	-	-	-	-
AntGPT [47]	13.3B*	246M*	0.700	0.717	0.658	0.655	0.650	0.650
PALM [19]	11.4B*	219M*	-	-	0.656	0.640	0.647	0.612
Video-LLaVA	303M	7.2B	0.697	0.711	-	-	-	-
Methods with <1B parameters								
RepLAI [27]	0	22M*	0.755	0.834	-	-	-	-
SlowFast [14]	63.4M	181M	0.745	0.779	-	-	0.717	0.736
VCLIP [7]	149M*	181M*	0.715	0.748	0.739	0.769	-	-
ICVAE [25]	63.4M*	55M*	-	-	0.741	0.740	-	-
HierVL [3]	0	192M	-	-	0.724	0.735	-	-
XCLIP-MultiHead [30]	575M	91M	0.736	0.761	-	-	-	-
LaViLa-MultiHead [48]	489M	91M	0.724	0.749	-	-	0.720	0.727
LaViLa-MHCAD (Ours)	489M	157M	**0.699**	**0.717**	**0.719**	**0.722**	**0.689**	**0.695**

5 Experiments

5.1 Implementation Details

We initialize the visual encoder and text encoders with pretrained vision-language weights from LaViLa-Large [48], providing strong semantic priors for both modalities. Following standard practice in prior work [14], we keep the visual encoder frozen during training to preserve the learned representations. Unlike many prior works [14] that pretrain on action recognition using the same anticipation videos before fine-tuning on the anticipation task, we train the decoder directly on the anticipation objective from scratch. This direct optimization approach avoids potential distribution mismatch between recognition and anticipation tasks while significantly reducing overall training time.

We extract features from 32 uniformly sampled frames per observed video clip. The decoder is trained for 50 epochs using the Adam optimizer [20], following the standard training protocol established by baseline methods [14]. We use a learning rate of 5×10^{-4} with a batch size of 16. For text encoder adaptation, we employ bottleneck adapters with dimension $d_{\text{bottleneck}} = d_e/4$, where d_e is the text embedding dimension. The decoder consists of $L_d = 2$ layers. During training, we apply DropPath with probability $p_{\text{drop}} = 0.1$ for regularization.

5.2 Main Results

Closed-Set Evaluation. Table 1 presents results on Ego4D across three evaluation sets. LLM/VLM-based methods (greyed out) are not directly comparable

Table 2: Open-Vocabulary Evaluation on EK100 and EGTEA+. The action metric combines verb and noun predictions into a joint metric. * indicates parameter counts estimated from paper details and/or code when not explicitly provided by authors.

Method	Frozen Params	Trained Params	EK100			EGTEA+		
			Verb ↓	Noun ↓	Action ↓	Verb ↓	Noun ↓	Action ↓
LLM/VLM Foundation Models with >7B parameters								
AntGPT [47]	13.3B*	246M*	0.933	0.958	0.993	0.961	0.974	0.998
Video-LLaVA [23]	303M	7.2B	0.925	0.945	0.991	0.959	0.970	0.994
Methods with Base Transformer Backbone								
XCLIP-MultiHead [30]	195M	66M	0.915	0.983	-	0.940	0.978	-
LaViLa-MultiHead [48]	152M	66M	0.889	0.981	-	0.931	0.970	-
LaViLa-MHCAD (Ours)	152M	76M	**0.779**	**0.960**	-	**0.903**	**0.950**	-
Methods with Large Transformer Backbone								
XCLIP-MultiHead [30]	575M	91M	0.868	0.971	0.985	0.912	0.966	0.981
LaViLa-MultiHead [48]	489M	91M	0.849	0.968	0.982	0.904	0.958	0.979
LaViLa-MHCAD (Ours)	489M	157M	**0.728**	**0.936**	**0.969**	**0.864**	**0.927**	**0.968**

as they rely on 7B+ parameter language models pretrained on massive text corpora, making them a fundamentally different class of approaches. Among methods with less than 1B parameter, our LaViLa-MHCAD outperforms all baselines, including our LaViLa-MultiHead and XCLIP-MultiHead baselines, which are constructed by applying prediction heads directly to CLS tokens without cross-attention aggregation, following [14]. Our method achieves 0.689/0.695 vs. 0.720/0.727 for LaViLa-MultiHead on Ego4D v2 test, confirming that cross-attention decoder aggregation provides consistent gains. We also surpass SlowFast [14], VCLIP [7], ICVAE [25], and HierVL [3] across all reported splits, while using only 489M frozen and 157M trainable parameters. Note that some baseline results are unavailable as the Ego4D test server for long-term anticipation is no longer active, and ground-truth test annotations are not publicly accessible.

Open-Vocabulary Evaluation. Table 2 evaluates cross-dataset generalization, where all models are trained exclusively on Ego4D and tested on EK100 and EGTEA+ without any fine-tuning. We report verb and noun errors, together with the action metric that combines verb and noun predictions into a joint metric. Although LLM/VLM methods (greyed out) are not fair comparisons due to their billion-scale parameters and large language model priors, our LaViLa-MHCAD still outperforms them on both datasets, achieving 0.728/0.936 on EK100 and 0.864/0.927 on EGTEA+ with the large backbone, compared to the best LLM/VLM results of 0.925/0.945 and 0.959/0.970, respectively. We extend the MultiHead baselines of [30, 48] to include both base and large backbone variants, providing a more comprehensive comparison across model scales. Our method substantially outperforms all MultiHead baselines, with the consistent margins across both backbone sizes directly demonstrating the benefit of our cross-attention decoder over independent per-horizon prediction heads for open-vocabulary long-term action anticipation.

Table 3: Open-vocabulary ablations on EK100 and EGTEA+. We ablate the decoder design and the position embedding. Lower is better.

Method	EK100		EGTEA+	
	Verb ↓	Noun ↓	Verb ↓	Noun ↓
Decoder design				
No self-attention	0.805	0.931	0.869	0.932
Non-causal self-attention	0.821	0.939	0.895	0.941
Swap self/cross-attention	0.769	0.937	0.886	0.941
LaViLa-FUTR	0.836	0.979	0.902	0.982
Position embedding				
DPE	0.746	0.962	0.885	0.952
Absolute position embedding	0.783	0.969	0.904	0.974
LaViLa-MHCAD (Ours)	0.728	0.936	0.864	0.927

6 Analysis and Ablations

To understand the contribution of each component in our approach, MHCAD, we conduct comprehensive ablation studies. All ablations are conducted using our full model configuration unless otherwise specified, with results reported on the Ego4D v2 test (closed-set) and EK100/EGTEA+ validation sets (open-vocabulary).

Decoder Design Ablations: We ablate our decoder design and position embedding in Tab. 3. For the decoder, removing causal self-attention among the horizon queries (*No self-attention*) slightly lowers noun error on EK100 but degrades verb error and all EGTEA+ metrics. Replacing the causal mask with bidirectional self-attention (*Non-causal self-attention*) or swapping the order of self-attention and cross-attention (*Swap self/cross-attention*) likewise underperforms our design, confirming that conditioning each query only on preceding queries is beneficial: future actions unfold sequentially, so a horizon query should not attend to actions that occur later. We further compare against *LaViLa-FUTR*, which pairs our frozen LaViLa encoder with the FUTR [13] decoder and uses cosine similarity to the text features for open-vocabulary prediction. It performs worse than ours, as it relies only on the final encoder representation with absolute 1D position embeddings and unconstrained self-attention over queries, whereas our decoder cross-attends to multiple encoder layers with factored 4D rotary embeddings and causal self-attention. For the position embedding, our factored 4D rotary embeddings outperform both dynamic positional encoding (DPE) and absolute position embeddings. Overall, our full model attains the best verb error on both datasets and the best noun error on EGTEA+.

Impact of Decoder Layers: As shown in Fig. 5, increasing decoder depth from 0 (no decoder) to 2 layers yields substantial improvements across all datasets, with notable gains on Ego4D v2 test (verb: 0.720→0.689, noun: 0.727→0.695),

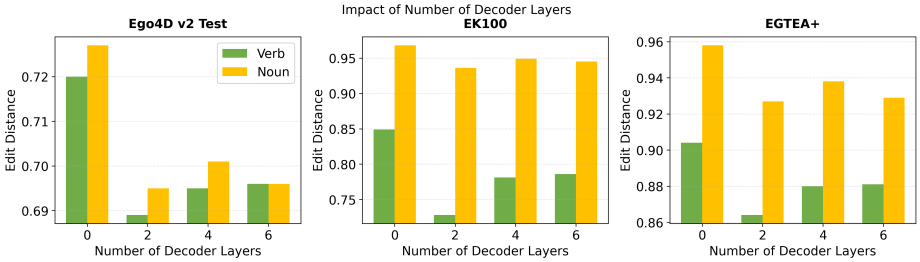

Fig. 5: Impact of decoder depth on performance across datasets: The plot shows that 2 layers provide an optimal balance between capacity and generalization.

Table 4: Ablation study on text adapter mechanisms.

Text Adapter	Ego4D v2 Test		EK100		EGTEA+	
	Verb ↓	Noun ↓	Verb ↓	Noun ↓	Verb ↓	Noun ↓
Adapter + Prompts	0.694	0.695	0.734	0.938	0.871	0.929
Prompts Only	0.697	0.695	0.736	0.937	0.876	0.929
Adapter Only	0.689	0.695	0.728	0.936	0.864	0.927

EK100 (verb: 0.849→0.728, noun: 0.968→0.936), and similarly for EGTEA+ (verb: 0.904→0.864, noun: 0.958→0.927). Using no decoder, *i.e.*, applying prediction heads directly to CLS tokens without cross-attention aggregation (equivalent to the LaViLa-MultiHead baseline in Tables 1 and 2), performs considerably worse, confirming the importance of explicit feature aggregation. Beyond 2 layers, performance degrades, suggesting that a shallow decoder best balances model capacity and cross-dataset generalization.

Text Adaptation Mechanisms: To enable open-vocabulary action recognition, we employ lightweight adaptation mechanisms for our frozen text encoders. We evaluate two approaches: (1) *Prompt tuning*, which prepends learnable prompt tokens to class name embeddings as soft conditioning signals [40], and (2) *Bottleneck adapters*, which insert learnable down-projection and up-projection layers after each transformer block [43]. Table 4 compares these strategies individually and in combination. Using bottleneck adapters alone achieves the best performance across all datasets, outperforming both the combination of adapters with prompt tuning and prompts alone. On EK100, adapter-only adaptation yields a verb edit distance of 0.728 compared to 0.734 with adapters and prompts, and 0.736 with prompts only. The trend holds consistently across the Ego4D v2 test and EGTEA+, with the adapter-only approach achieving 0.689/0.695 and 0.864/0.927 for verbs and nouns, respectively. This suggests that bottleneck adapters provide sufficient task-specific adaptation capacity without the additional complexity of learnable prompts. Prompts alone perform the worst, indicating that modifying only the input embeddings is insufficient for adapting the text encoder to the anticipation task. At the same time, adapters inserted within the transformer layers can more effectively modulate the representations throughout the network depth.

Table 5: Ablation study on the number of queries and head style.

Num Queries	Head Style	Ego4D v2 Test		EK100		EGTEA+	
		Verb ↓	Noun ↓	Verb ↓	Noun ↓	Verb ↓	Noun ↓
Single	Per Horizon	0.719	0.727	0.759	0.992	0.901	0.970
20	Per Horizon	0.701	0.705	0.837	0.947	0.879	0.946
20	Shared	0.689	0.695	0.728	0.936	0.864	0.927

Query Design and Prediction Heads: Table 5 examines the impact of using multiple horizon-specific queries versus a single query, and shared versus per-horizon prediction heads. Using a single query with per-horizon heads (row 1) performs worst across all datasets, achieving 0.719/0.727 on the Ego4D v2 test set, demonstrating that a single aggregated representation cannot effectively capture the diverse temporal dynamics across multiple future horizons. Introducing 20 horizon-specific queries with per-horizon heads (row 2) improves performance (0.701/0.705), but the best results are achieved with 20 queries and shared prediction heads (row 3: 0.689/0.695). This pattern is consistent across EK100 and EGTEA+, with particularly large improvements on EK100 (verb: 0.759→0.837→0.728). The superiority of shared heads suggests that while horizon-specific queries are essential for learning distinct temporal patterns, using separate prediction heads for each horizon limits the generalization for the open-vocabulary evaluation. More ablations are in the supplementary material.

7 Conclusion

We introduced the first open-vocabulary cross-dataset evaluation framework for action anticipation, addressing a critical gap in current evaluation practices that focus exclusively on closed-set settings. By training on Ego4D and testing on EPIC-Kitchens and EGTEA+, our framework reveals substantial differences in model generalization to novel action vocabularies. Our Multi-Horizon Cross-Attention Decoder demonstrates significant improvements in this open-vocabulary setting, achieving up to 20% relative gains over the baselines while using far fewer parameters than billion-scale models. While our approach is a first step toward open-vocabulary long-term anticipation, we expect that our evaluation protocol will encourage future advancements in this new research area.

Acknowledgements

The work has been supported by the Federal Ministry of Research, Technology, and Space (BMFTR) under grant no. 16IS22094A WEST-AI and the ERC Consolidator Grant FORHUE (101044724). For the computations involved in this research, we acknowledge EuroHPC Joint Undertaking for awarding us access to Leonardo at CINECA, Italy, through EuroHPC Regular Access Call - proposal No. EHPC-REG-2025R01-218.

References

1. Abu Farha, Y., Gall, J.: Uncertainty-aware anticipation of activities. In: ICCVW (2019)
2. Abu Farha, Y., Richard, A., Gall, J.: When will you do what?-Anticipating temporal occurrences of activities. In: CVPR (2018)
3. Ashutosh, K., Girdhar, R., Torresani, L., Grauman, K.: Hiervl: Learning hierarchical video-language embeddings. In: CVPR (2023)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
5. Chen, G., Zheng, Y.D., Wang, J., Xu, J., Huang, Y., Pan, J., Wang, Y., Wang, Y., Qiao, Y., Lu, T., et al.: Videollm: Modeling video sequence with large language models. arXiv preprint, arXiv:2305.13292 (2023)
6. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. IJCV (2022)
7. Das, S., Ryoo, M.S.: Video+ clip baseline for ego4d long-term action anticipation. arXiv preprint, arXiv:2207.00579 (2022)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
9. Furnari, A., Farinella, G.M.: Rolling-unrolling lstms for action anticipation from first-person video. TPAMI (2020)
10. Girase, H., Agarwal, N., Choi, C., Mangalam, K.: Latency matters: Real-time action forecasting transformer. In: CVPR (2023)
11. Girdhar, R., Grauman, K.: Anticipative video transformer. In: ICCV (2021)
12. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: CVPR (2017)
13. Gong, D., Lee, J., Kim, M., Ha, S., Cho, M.: Future transformer for long-term action anticipation. In: CVPR (2022)
14. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: CVPR (2022)
15. Guo, H., Agarwal, N., Lo, S.Y., Lee, K., Ji, Q.: Uncertainty-aware action decoupling transformer for action anticipation. In: CVPR (2024)
16. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: CVPR (2019)
17. Hussein, N., Gavves, E., Smeulders, A.W.: Videograph: Recognizing minutes-long human activities in videos. arXiv preprint, arXiv:1905.05143 (2019)
18. Ke, Q., Fritz, M., Schiele, B.: Time-conditioned action anticipation in one shot. In: CVPR (2019)
19. Kim, S., Huang, D., Xian, Y., Hilliges, O., Van Gool, L., Wang, X.: Palm: Predicting actions through language models. In: ECCV (2024)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
21. Lai, B., Toyer, S., Nagarajan, T., Girdhar, R., Zha, S., Rehg, J.M., Kitani, K., Grauman, K., Desai, R., Liu, M.: Human action anticipation: A survey. arXiv preprint, arXiv:2410.14045 (2024)
22. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. In: ECCV (2018)

23. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: EMNLP (2024)
24. Lin, K.Q., Wang, J., Soldan, M., Wray, M., Yan, R., Xu, E.Z., Gao, D., Tu, R.C., Zhao, W., Kong, W., et al.: Egocentric video-language pretraining. In: NeurIPS (2022)
25. Mascaro, E.V., Ahn, H., Lee, D.: Intention-conditioned long-term human egocentric action anticipation. In: WACV (2023)
26. Mittal, H., Agarwal, N., Lo, S.Y., Lee, K.: Can't make an omelette without breaking some eggs: Plausible action anticipation using large video-language models. In: CVPR (2024)
27. Mittal, H., Morgado, P., Jain, U., Gupta, A.: Learning state-aware visual representations from audible interactions. NeurIPS (2022)
28. Nagarajan, T., Li, Y., Feichtenhofer, C., Grauman, K.: Ego-topo: Environment affordances from egocentric video. In: CVPR (2020)
29. Nawhal, M., Jyothi, A.A., Mori, G.: Rethinking learning approaches for long-term action anticipation. In: ECCV (2022)
30. Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., Ling, H.: Expanding language-image pretrained models for general video recognition. In: ECCV (2022)
31. Pramanick, S., Song, Y., Nag, S., Lin, K.Q., Shah, H., Shou, M.Z., Chellappa, R., Zhang, P.: Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In: ICCV (2023)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
33. Roy, D., Rajendiran, R., Fernando, B.: Interaction region visual transformer for egocentric action anticipation. In: WACV (2024)
34. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: ECCV (2020)
35. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing (2024)
36. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint, arXiv:2307.09288 (2023)
37. Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint, arXiv:2109.08472 (2021)
38. Wang, Y., Li, K., Li, X., Yu, J., He, Y., Chen, G., Pei, B., Zheng, R., Wang, Z., Shi, Y., et al.: Internvideo2: Scaling foundation models for multimodal video understanding. In: ECCV (2024)
39. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint, arXiv:2212.03191 (2022)
40. Wasim, S.T., Naseer, M., Khan, S., Khan, F.S., Shah, M.: Vita-clip: Video and text adaptive clip via multimodal prompting. In: CVPR (2023)
41. Wasim, S.T., Suleman, H., Zatsarynna, O., Naseer, M., Gall, J.: Mixant: Observation-dependent memory propagation for stochastic dense action anticipation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14613–14622 (2025)
42. Wu, C.Y., Li, Y., Mangalam, K., Fan, H., Xiong, B., Malik, J., Feichtenhofer, C.: Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In: CVPR (2022)

43. Yang, T., Zhu, Y., Xie, Y., Zhang, A., Chen, C., Li, M.: Aim: Adapting image models for efficient video action recognition. In: ICLR (2023)
44. Zatsarynna, O., Bahrami, E., Abu Farha, Y., Francesca, G., Gall, J.: Gated temporal diffusion for stochastic long-term dense anticipation. ECCV (2024)
45. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023)
46. Zhao, H., Wildes, R.P.: On diverse asynchronous activity anticipation. In: ECCV (2020)
47. Zhao, Q., Wang, S., Zhang, C., Fu, C., Do, M.Q., Agarwal, N., Lee, K., Sun, C.: Antgpt: Can large language models help long-term action anticipation from videos? arXiv preprint, arXiv:2307.16368 (2023)
48. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: CVPR (2023)
49. Zhong, Z., Martin, M., Schneider, D., Lerch, D.J., Wu, C., Diederichs, F., Gall, J., Beyerer, J.: Scalable video action anticipation with cross linear attentive memory. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 8113–8123 (March 2026)
50. Zhong, Z., Martin, M., Voit, M., Gall, J., Beyerer, J.: A survey on deep learning techniques for action anticipation. arXiv preprint, arXiv:2309.17257 (2023)
51. Zhong, Z., Schneider, D., Voit, M., Stiefelhagen, R., Beyerer, J.: Anticipative feature fusion transformer for multi-modal action anticipation. In: WACV (2023)
52. Zhong, Z., Wu, C., Martin, M., Voit, M., Gall, J., Beyerer, J.: Diffant: Diffusion models for action anticipation. arXiv preprint, arXiv:2311.15991 (2023)