SpotEM: Efficient Video Search for Episodic Memory

Santhosh Kumar Ramakrishnan [1] **Ziad Al-Halah** [2] **Kristen Grauman** [1 3]

Abstract

The goal in episodic memory (EM) is to search a long egocentric video to answer a natural language query (e.g., *"where did I leave my purse?"*). Existing EM methods exhaustively extract expensive fixed-length clip features to look everywhere in the video for the answer, which is infeasible for long wearable-camera videos that span hours or even days. We propose SpotEM, an approach to achieve efficiency for a given EM method while maintaining good accuracy. SpotEM consists of three key ideas: 1) a novel clip selector that learns to identify promising video regions to search conditioned on the language query; 2) a set of low-cost semantic indexing features that capture the context of rooms, objects, and interactions that suggest where to look; and 3) distillation losses that address the optimization issues arising from end-to-end joint training of the clip selector and EM model. Our experiments on 200+ hours of video from the Ego4D EM Natural Language Queries benchmark and three different EM models demonstrate the effectiveness of our approach: computing only 10% – 25% of the clip features, we preserve 84% – 97% of the original EM model's accuracy. Project page: https://vision.cs.utexas.edu/projects/spotem

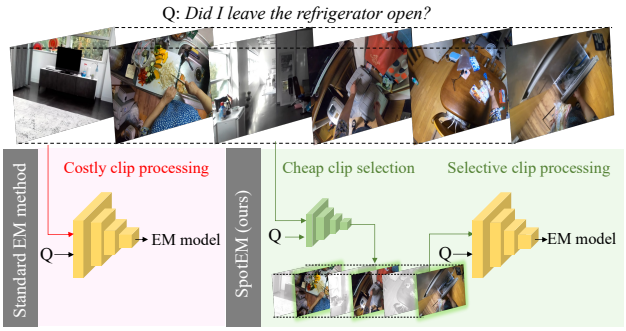

Figure 1. Standard EM methods (see left) divide videos into fixed-length clips and perform costly processing of every clip. The processed clips are provided to the EM model for query-conditioned search. We propose SpotEM, a clip-selection approach that spots clips relevant to the query cheaply, and selectively processes these clips to serve as inputs to the EM model (see right).

1. Introduction

The limitations of human memory can pose an obstacle for our day-to-day activities. We forget where we put things (*"where did I leave my car keys?"*), fail to notice the state of particular objects (*"did I turn off the stove? how much milk is left in the fridge?"*), and struggle to recall details about past events (*"who did I run into while jogging last week? what was the name of the bakery where we bought the muffins?"*). First-person or "egocentric" perception from wearable devices like augmented reality (AR) glasses could remove that cognitive load, allowing a user to ask questions on-the-fly about their own past visual experience. Similarly, an intelligent mobile robot could relay valuable information about what it has seen based on people's conversational queries (*"did anyone feed the dog yet?"*).

This vision of a *personal episodic memory* prompts interesting new challenges in computer vision and multimodal learning. Not only does it require recognizing objects and activities from the first-person perspective, but also identifying which visual content is sufficient to answer a question expressed in free-form natural language, accounting for the fact that the correct answer may occupy only a tiny portion of the entire video. Moreover, all this must be done in a scalable manner, given that the visual history of a user will ultimately span hours, days, weeks, or more.

The recently introduced Episodic Memory (EM) benchmark[1] from Ego4D targets this task (Grauman et al., 2022): given a natural language query and a long egocentric video, identify the precise temporal window containing the answer. The EM benchmark has attracted significant attention and is the subject of community-wide competitions (Grauman et al., 2022; Liu et al., 2022; Lin et al., 2022b; Hou et al., 2022; Mo et al., 2022; Chen et al., 2022) attracting dozens

[1]UT Austin [2]University of Utah [3]FAIR, Meta AI. Correspondence to: S. Ramakrishnan <sramakrishnan@utexas.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]In particular, we focus on Ego4D's NLQ, the EM variant with natural language queries, as opposed to other variants defined with object or activity queries.

of submissions in the first year alone. Hallmarks of the benchmark are the free-form nature of the natural language queries, the long-form nature of the egocentric videos (8.2 minutes on average), and the short responses that span a tiny fraction of the overall video (2% on average). The result is a "needle-in-the-haystack" problem for video localization. Importantly, these facets set the EM task apart from traditional activity recognition and language grounding tasks, which feature much shorter videos in which most content is relevant (Caba Heilbron et al., 2015; Jiang et al., 2018; Gao et al., 2017) (e.g., 30 second video in which a third of it displays the target action).

While current EM methods have made exciting headway, they neglect the practical scaling issue that is so central to episodic memory. Today's methods extract expensive spatio-temporal features for densely sampled clips throughout the video (Feichtenhofer et al., 2019; Bertasius et al., 2021; Tong et al., 2022; Girdhar et al., 2022). This step alone contributes over 99.9% of an EM model's computational cost.[2] Such an approach becomes intractable as the video length grows, especially for real-time applications like AR and robotics where the constrained on-board computation severely limits the ability to operate such heavy-weight models. Besides, EM is just one of several functions that need to be supported by these devices—all resources cannot be devoted to just computing features for EM.

We observe that 1) not all parts of the video are useful for reasoning about a given query, and 2) there are high-level visual semantics about rooms, objects and interactions that could steer our attention towards where to look. For example, given the queries, *"did I leave the lights on in the living room?"* or *"did I close the refrigerator?"*, we can ignore video clips recorded in rooms other than the living room or kitchen, respectively. Notably, these associations cannot be neatly enumerated, however, given the free-form nature of the queries. In the query *"who was I with when I first boarded the golf cart?"*, the chain of reasoning becomes much more complex since we need to reason about all instances of boarding a golf cart, and identify those instances where there was someone else with you. This points to the need for learning query-conditioned priors that can use such high-level semantics to help narrow down the search task.

We build upon these intuitions to propose SpotEM, a novel approach to make a given EM method efficient. See Figure 1. The idea is to preview the video using cheap indexing features, intelligently select a small subset of *query-relevant* clips, and only use these clips for the full EM search. This can cut down computational costs without sacrificing model performance. Our approach differs from prior work for efficient action recognition (Wu et al., 2018; Yeung et al., 2016; Wu et al., 2019a; Feichtenhofer, 2020; Gao et al., 2020) and video captioning (Chen et al., 2018; Suin & Rajagopalan, 2020) that assumes much shorter inputs and does not condition on free-form language queries.

To tackle this challenging setting with long-form egocentric videos and language queries, we design *MemorySpotter*, a novel clip selection architecture that uses a cross-modal transformer to recursively preview the video and identify query-relevant clips. We further design a set of inexpensive semantic-indexing features that capture video context about **r**ooms/scenes visited, **i**nteractions observed, and the **o**bjects present (*RIO*), thereby exposing to the model the high-level visual context that may support the free-form query text. Further, we propose expert-based distillation losses to address optimization issues arising from jointly training MemorySpotter and the EM modules. Our experiments on the Ego4D EM benchmark demonstrate that our approach is effective and versatile: when tested with multiple EM methods, it achieves 95+% of the original EM method's performance while computing heavy-weight video features for only 25% of the clips. We also perform ablation studies to validate the design of SpotEM and present a detailed study of its clip selection behaviors.

2. Related work

Egocentric video understanding. Unlike internet-style data, egocentric video captures the camera wearer's perspective of their activities, and is the subject of multiple datasets (Fathi et al., 2011; Kazakos et al., 2019; Furnari & Farinella, 2020; Damen et al., 2022; Grauman et al., 2022). Ego-video raises new research problems in human-object interaction (Cai et al., 2018; Damen et al., 2014), activity recognition (Kazakos et al., 2019; Zhou & Berg, 2015), anticipation (Abu Farha et al., 2018; Girdhar & Grauman, 2021), video summarization (Del Molino et al., 2016; Lee & Grauman, 2015), and spatial organization (Ortis et al., 2017; Furnari et al., 2016; Nagarajan et al., 2020; Price et al., 2022). We focus on the recently introduced episodic memory task, which requires answering queries about long-form egocentric videos (Grauman et al., 2022).

Episodic memory. The goal in episodic memory is to temporally localize the response to natural language queries. This task extends video question answering (Xu et al., 2017; Rohrbach et al., 2017; Xu et al., 2021; Zhang et al., 2020a) to the challenging egocentric setting. The EM benchmark offers a compelling step towards episodic memory applications in AR and has been the subject of multiple challenges at top conferences (Grauman et al., 2022).[3] Prior

[2] A single clip feature from Chen et al. (2022) consumes 2.09 TFLOPs, while EM modules from Zhang et al. (2020a) (excluding the video backbone) consume 3 GFLOPs for the whole video.

[3] CVPR 2022: https://ego4d-data.org/workshops/cvpr22/
ECCV 2022: https://ego4d-data.org/workshops/eccv22/

work adapts video-language grounding methods such as 2D-TAN (Zhang et al., 2020b), VSLNet (Zhang et al., 2020a), and VSLNet-L (Zhang et al., 2021) to perform EM search, and the state-of-the-art methods develop video representations (Lin et al., 2022b; Chen et al., 2022) and data augmentation strategies (Liu et al., 2022). Unlike prior work which is purely motivated by accuracy, we tackle the orthogonal problem of search efficiency, which is critical for real-world EM applications. Our results show SpotEM's versatility: deployed to augment three popular EM models (Chen et al., 2022; Liu et al., 2022; Lin et al., 2022b), it successfully achieves substantial efficiency gains for each one.

Efficient video models. Prior work develops efficient video recognition architectures through compute-efficient modules (Feichtenhofer, 2020), feature-distillation (Zhang et al., 2016; Gao et al., 2020), compressed video processing (Zhang et al., 2016; Wu et al., 2018), or adaptively choosing between cheap and expensive inputs and modules (Zhu et al., 2020; Meng et al., 2020; Li et al., 2021). Orthogonal to this are frame-sampling methods that select informative subsets of the video for video recognition (Chen et al., 2011; Yeung et al., 2016; Wu et al., 2019a), typically by first previewing the video with an inexpensive module. For example, Chen et al. (2011) use an inexpensive background subtraction module to filter out irrelevant frames, while Korbar et al. (2019) predict each clip's saliency and select the most salient clips for processing. Alternatively, reinforcement learning (RL) can be used to learn frame-selection policies for action recognition and detection (Yeung et al., 2016; Fan et al., 2018). Wu et al. (2019a) improve over RL methods by using the Gumbel-Softmax trick to reduce frame selection to a supervised learning problem. Some methods further leverage audio to guide the sampling process (Jiang et al., 2015; Gao et al., 2020; Panda et al., 2021).

Rather than sequentially processing one clip at a time (Wu et al., 2019b; Yeung et al., 2016; Wu et al., 2019a), our clip selection architecture simultaneously previews all of the video at once using cheap image features and makes selection choices among all clips. Furthermore, prior models that do parallel sampling preview the entire video only once to make sampling decisions (Korbar et al., 2019; Lin et al., 2022a). We instead propose a *recursive* approach that previews the video multiple times to actively grow the set of observed clips. At each recursive step, we select a subset of clips, extract their (heavier) features, and incorporate this knowledge for future clip selection steps. We demonstrate the significance of these contributions in our experiments. Beyond architectural improvements over prior work, SpotEM also has novel contributions of designing a) semantic index features relevant to EM and b) distillation losses to address optimization limitations arising from jointly training a clip selector and the task models.

3. Approach

We propose SpotEM, a clip-selection approach that intelligently spots query-relevant clips for efficient EM. Our overall approach consists of a novel clip selection architecture called MemorySpotter, our RIO semantic indexing features to select clips relevant for EM, and distillation losses to address optimization issues arising from jointly training MemorySpotter with EM task modules. MemorySpotter previews the video using our RIO features, which are obtained by selecting a single image from each clip and encoding them using efficient image encoders (Tan & Le, 2019). Heavy clip features are then extracted from only the smaller subset of clips selected by MemorySpotter. Next, we review the EM task definition and discuss our approach.

3.1. Episodic memory task

The goal in natural-language episodic memory is to perform query-driven reasoning about long-form egocentric videos (Grauman et al., 2022). Formally, given an egocentric video $\mathcal{V}$ capturing a camera wearer's past experiences and a query text $\mathcal{Q}$, the task requires temporally localizing where the answer can be seen in the video, i.e., a response window $\mathcal{R} = [t_s, t_e]$ defined by start t_s and end t_e.

Recent work adapts video-language grounding models like VSLNet (Zhang et al., 2020a) and VSLNet-L (Zhang et al., 2021) to achieve state-of-the-art results for EM (Lin et al., 2022b; Liu et al., 2022; Chen et al., 2022). Our analysis shows that these methods devote over 99.9% of their computation to extracting spatio-temporal features for fixed-length clips that are sampled densely from the video. We instead propose to preview the video cheaply and identify query-relevant clips to perform EM efficiently.

3.2. SpotEM for efficient episodic memory

We now provide an overview of our SpotEM approach. See Figure 2. The model inputs consist of the video $\mathcal{V}$ with clips $[\mathcal{V}_1, \cdots, \mathcal{V}_L]$, and a text query $\mathcal{Q}$ with words $[\mathcal{Q}_1, \cdots, \mathcal{Q}_T]$. First, a pretrained semantic indexer, which consists of one or more image encoders, is used to extract semantic index features $\boldsymbol{s} \in \mathbb{R}^{L \times D_s}$.

$$\boldsymbol{s} = [s_1, s_2, \cdots, s_L] = \text{SemanticIndexer}(\mathcal{V}). \quad (1)$$

These are image features extracted by sampling one image within each video clip, and they are inexpensive to compute (2-3 orders lower cost than clip features). These will serve as an initial preview of the video for intelligent clip selection (more details in Section 3.4).

A pretrained DistillBERT backbone (Sanh et al., 2019) is then used to extract the query text features $\boldsymbol{q} \in \mathbb{R}^{T \times D_q}$.

$$\boldsymbol{q} = [q_1, q_2, \cdots, q_T] = \text{TextBackbone}(Q). \quad (2)$$

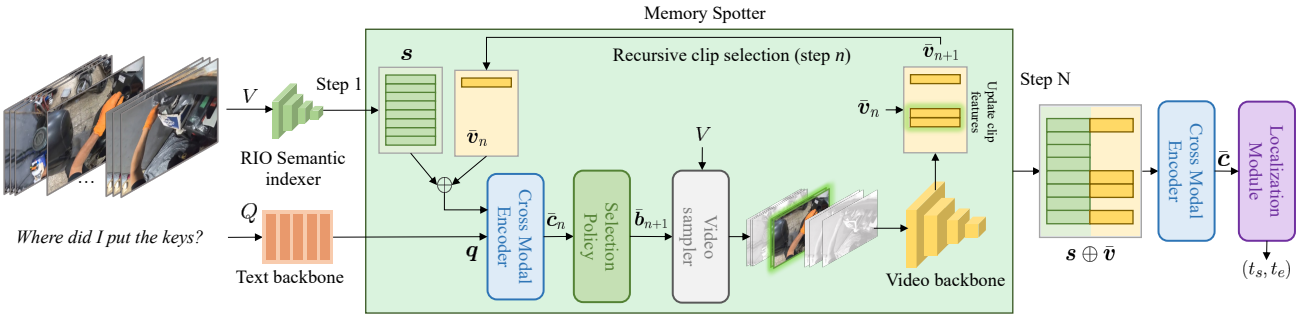

Figure 2. **SpotEM for efficient video search in episodic memory**: We enhance standard EM models with the ability to intelligently select clips for expensive feature extraction, with the aim of reducing computational cost without affecting accuracy. First, we extract our RIO semantic indexing features $\boldsymbol{s}$ that capture room, interaction, and object context in the video using a cheap semantic indexer. We then encode the query using a text backbone to obtain query features $\boldsymbol{q}$ (see left). The MemorySpotter module operates recursively to select a subset of clips relevant to the query (see center). It first previews the video using semantic index $\boldsymbol{s}$. It then alternates between selecting subsets of video clips $\bar{\mathcal{V}}_n$ for expensive feature extraction, and previewing the video again with both the semantic index s and the previously selected clip features $\bar{\boldsymbol{v}}_{n+1}$. After N steps of the MemorySpotter, we use its cumulative selected clip features $\bar{\boldsymbol{v}} = \bar{\boldsymbol{v}}_{N+1}$ and semantic index $\boldsymbol{s}$ to perform EM (see right). A cross-modal encoder jointly reasons about the concatenated features $\boldsymbol{s} \oplus \bar{\boldsymbol{v}}$ and $\boldsymbol{q}$, to obtain a cross-modal embedding $\bar{\boldsymbol{c}}$. A localization module then predicts the temporal extents of the response. Note that the cross-modal encoder inside MemorySpotter shares weights with the cross-modal encoder on the right. At step $n = 1$, no clips are selected, i.e., $\bar{\boldsymbol{v}}_1 = \vec{\mathbf{0}}$.

The MemorySpotter module uses the semantic index $\boldsymbol{s}$, query features $\boldsymbol{q}$, and the video $\mathcal{V}$ to recursively identify a final subset of video clips $\bar{\mathcal{V}}$ that are relevant to the query. The expensive clip features for $\bar{\mathcal{V}}$ are obtained using pre-trained video backbones like Timesformer (Bertasius et al., 2021) and VideoMAE (Tong et al., 2022) (see Sec. 4.1).[4]

$$\bar{\boldsymbol{v}} = \text{VideoBackbone}(\bar{\mathcal{V}}) \in \mathbb{R}^{L \times D_v} \tag{3}$$

A cross-modal encoder uses the concatenated clip and semantic index features $\boldsymbol{s} \oplus \bar{\boldsymbol{v}}$ and the query features $\boldsymbol{q}$ to perform cross-modal reasoning to enhance the video features with query-specific information.

$$\bar{\boldsymbol{c}} = \text{CrossModalEncoder}(\boldsymbol{s} \oplus \bar{\boldsymbol{v}}, \boldsymbol{q}) \in \mathbb{R}^{L \times D_h} \tag{4}$$

Finally, a localization module is used to predict the temporal extent of the response $\hat{\mathcal{R}}$ based on $\bar{\boldsymbol{c}}$:

$$\hat{\mathcal{R}} = [\hat{t}_s, \hat{t}_e] = \text{LocalizationModule}(\bar{\boldsymbol{c}}). \tag{5}$$

This formulation augments existing EM methods to be efficient. Specifically, EM methods like VSLNet (Zhang et al., 2020a) and VSLNet-L (Zhang et al., 2021) use video backbones to extract clip features, a text backbone to extract query features, a cross-model encoder to jointly encode the two modalities, and a localization module to predict the response locations (see Figure 6). SpotEM modulates the video inputs to these EM models by intelligently selecting a subset of clips for heavy clip feature extraction. While the architectural details of the cross-modal encoder and localization modules vary across EM methods (described in Appendix A), our approach remains unchanged.

[4]The video features for clips not selected are set to zeros in $\bar{\boldsymbol{v}}$.

SpotEM achieves efficiency by previewing the video cheaply using the semantic indexer (detailed in Sec. 3.4) and then recursively selecting a subset of clips relevant to the query using MemorySpotter, which we describe next.

3.3. MemorySpotter architecture description

One of our key contributions is the MemorySpotter architecture for intelligent clip selection for EM. See Figure 2 (center). It first previews the entire video using low-cost semantic index ($\boldsymbol{s}$). It then alternates between selecting video clips for expensive feature extraction, and previewing the video again with both the semantic indexing features *and* the previously selected clip features ($\bar{\boldsymbol{v}}_n$).

Specifically, let $n \in [1, \cdots, N]$ denote the recursive step and $\bar{\boldsymbol{v}}_n \in \mathbb{R}^{L \times D_v}$ denote the clip features for the subset of video clips selected from steps 1 to $n-1$, where $\bar{\boldsymbol{v}}_1 = [0]_{L \times D_v}$ is a matrix of zeros. The semantic indexing features $\boldsymbol{s}$ are computed once before step 1 and kept fixed. At step n, the clip features $\bar{\boldsymbol{v}}_n$ are concatenated with $\boldsymbol{s}$ along the feature dimension to get $\boldsymbol{s} \oplus \bar{\boldsymbol{v}}_n \in \mathbb{R}^{L \times (D_v + D_s)}$. The concatenated visual features $\boldsymbol{s} \oplus \bar{\boldsymbol{v}}_n$ and $\boldsymbol{q}$ are used by the cross-modal encoder to perform joint reasoning:

$$\bar{\boldsymbol{c}}_n = \text{CrossModalEncoder}(\boldsymbol{s} \oplus \bar{\boldsymbol{v}}_n, \boldsymbol{q}). \tag{6}$$

The selection policy is a two-layered MLP that predicts a binary value (i.e., thresholded probabilities) per clip indicating whether clip features ought to be computed or not:[5]

$$\bar{\boldsymbol{b}}_{n+1} = \text{SelectionPolicy}(\bar{\boldsymbol{c}}_n) \in \{0, 1\}^L. \tag{7}$$

[5]Previously selected clips are excluded from predictions.

Finally, the video backbone is used to extract clip features for the clips selected in $\bar{\boldsymbol{b}}_{n+1}$. They are added to $\bar{\boldsymbol{v}}_n$ to get the updated clip features $\bar{\boldsymbol{v}}_{n+1}$. This selection process is repeated for N steps, and the cumulative set of clip features ($\bar{\boldsymbol{v}} = \bar{\boldsymbol{v}}_{N+1}$) is used to predict the EM response as shown in Equations (4) and (5).

3.4. RIO features for semantic indexing

Our SpotEM relies on cheap semantic indexing to preview the video and select clips intelligently. Prior work on efficient video recognition uses ImageNet-pretrained features for this purpose (Wu et al., 2019b;a; Gowda et al., 2021). However, ImageNet features capture only object-level information, which is insufficient for query-conditioned indexing in EM, where contextual cues about human-object interactions and room-level characteristics are needed. For example, object interaction features may be useful for the query, *"What tool did I use for fixing this part of the bike last time?"* while room-level features may be useful for the query, *"Did I leave the lights on in the living room?"* Hence, we design a set of low-cost semantic indexing features that capture context from the **r**ooms/scenes visited, human-object **i**nteractions, and the visible **o**bjects, which we term *RIO*. For each feature, we train an EfficientNet-b0 image encoder, which incurs 2-3 orders lower computational cost than typical clip encoders (Tan & Le, 2019).

Room features: We train an image encoder as a room/scene classifier to capture scene characteristics using scene annotations for Ego4D videos (Nagarajan et al., 2022). This includes 28 categories of both indoor scenes (e.g., bedroom, living room) and outdoor scenes (e.g., garden, porch).

Interaction features: We capture human-object interactions by training another image encoder using the contrastive vision-text pretraining objective from EgoVLP (Lin et al., 2022b). The objective is to maximize feature similarity between *video frames* and Ego4D's time-synchronized textual narrations reporting every step of the camera-wearer's activity (Grauman et al., 2022). In addition to image-text contrastive losses, we add losses to distill the clip-level features from a pretrained EgoVLP TimeSformer (Bertasius et al., 2021) backbone into the image encoder.

Object features: We replace the ImageNet features from prior work with self-supervised VICReg (Bardes et al., 2022) features trained on Ego4D images. The objective in VICReg is to learn image representations in a self-supervised way by minimizing reconstruction errors between two different views of an image (invariance), while maintaining diversity over each feature dimension (variance), and decorrelating pairs of feature dimensions (covariance). This captures object properties and also bridges the visual domain gap experienced by ImageNet features.

Overall, we sample one image within each video clip, extract each of the RIO features, and concatenate them to obtain the semantic indexing features $\boldsymbol{s}$ described in Figure 2.

3.5. Model optimization

Our SpotEM approach uses a MemorySpotter module for clip selection in conjunction with the EM modules, i.e., the cross-modal encoder and the localization module (see Figure 2). We jointly optimize these modules end-to-end on the EM task for improving task performance while only sampling a subset of video clips. We keep the video, semantic index, and text backbones frozen during training. Our loss function consists of the following terms: an EM task loss $\mathcal{L}_{\text{EM}}$, a selection loss $\mathcal{L}_{\text{SEL}}$, and two novel distillation losses for feature distillation $\mathcal{L}_{\text{FD}}$ and prediction distillation $\mathcal{L}_{\text{PD}}$.

EM task loss ($\mathcal{L}_{\text{EM}}$) optimizes the model to improve the EM NLQ performance. It typically consists of prediction losses for start and end locations, and a loss to prioritize clips overlapping with the response (Zhang et al., 2020a). The details are in Appendix B.

Selection loss ($\mathcal{L}_{\text{SEL}}$) optimizes the model to select a specified budget of clips and penalizes under-/over-sampling:

$$\mathcal{L}_{\text{SEL}} = \left(\mathbb{E}_{(\mathcal{V},\mathcal{Q})\sim D_{\text{train}}} \left[\frac{1}{L} \sum_{l=1}^{L} \bar{\boldsymbol{b}}^{l}_{\text{joint}} \right] - \gamma \right)^2, \quad (8)$$

where D_{train} is the training dataset and $\bar{\boldsymbol{b}}_{\text{joint}} = \sum_{n=1}^{N+1} \bar{\boldsymbol{b}}_n$ is the overall binary selections after N steps (refer to Equation (7)). $\mathcal{L}_{\text{SEL}}$ limits the fraction of clips selected, in expectation, to a predefined hyperparameter γ. This is similar to binary selection losses used in prior work (Wu et al., 2019a; Meng et al., 2020). We further regularize the per-step selection $\bar{\boldsymbol{b}}^{l}_{n}$ by encouraging the model to select $(\frac{\gamma}{N})L$ clips in each step from 1 to N. We ignore this term from Equation (8) for brevity. We observed this simple regularization can improve training stability. However, we did not explore more complex schemes that vary the number of selections conditioned on the iteration. Since MemorySpotter predicts binary selection values in Equation (7), it is not differentiable for gradient-based optimization. Following prior work, we use the Gumbel-Softmax trick to reparameterize argmax sampling using a softmax relaxation during training (Hazan & Jaakkola, 2012; Maddison et al., 2017; Jang et al., 2017; Wu et al., 2019a). See Appendix C for details.

Distillation losses: We observed some optimization difficulties during the joint optimization of MemorySpotter and EM modules (i.e., cross-modal encoder and localization module). Intuitively, the training of the EM modules is disrupted by the noisy clip-selection from MemorySpotter during the initial stage of learning. This in turn affects the optimization of MemorySpotter since it gets noisy gradients from the

Row	Clip selection method	η	Sem. index	MR@1	MR@5	TFLOPs
	ZeroClips	100	ImageNet	3.85	8.64	0.1
1	Random	90	ImageNet	4.35	9.85	26.8
2	Uniform	90	ImageNet	4.32	9.89	26.8
3	LiteEval (Wu et al., 2019a)	90	ImageNet	5.82	11.53	26.6
4	OCSampler (Lin et al., 2022a)	90	ImageNet	5.72	12.32	26.8
5	SpotEM w/o distill	91	ImageNet	6.99	12.95	24.8
6	SpotEM w/o distill	90	RIO	7.48	14.82	26.9
7	SpotEM (ours)	90	RIO	**9.62**	**17.07**	27.0
1	Random	75	ImageNet	4.91	11.03	67.0
2	Uniform	75	ImageNet	5.62	12.08	67.0
3	LiteEval (Wu et al., 2019a)	75	ImageNet	7.12	13.27	66.0
4	OCSampler (Lin et al., 2022a)	75	ImageNet	7.34	14.49	67.0
5	SpotEM w/o distill	76	ImageNet	9.22	16.90	65.2
6	SpotEM w/o distill	75	RIO	9.92	17.27	66.9
7	SpotEM (ours)	76	RIO	**11.06**	**18.92**	64.9
1	Random	50	ImageNet	7.52	14.85	133.9
2	Uniform	50	ImageNet	8.24	15.75	133.9
3	LiteEval (Wu et al., 2019a)	50	ImageNet	8.70	16.21	132.8
4	OCSampler (Lin et al., 2022a)	50	ImageNet	9.11	16.93	133.9
5	SpotEM w/o distill	53	ImageNet	9.35	17.18	125.5
6	SpotEM w/o distill	52	RIO	9.84	18.70	129.5
7	SpotEM (ours)	51	RIO	**11.56**	**19.90**	131.9
	AllClips (Chen et al., 2022)	0	-	11.45	20.56	267.6

Table 1. Comparing clip selection methods for the InternVideo EM method (Chen et al., 2022) on the Ego4D NLQ benchmark.

remaining modules. To break this negative feedback loop, we adopt a two-stage training pipeline. First, we train an expert EM model to perform the task without MemorySpotter (but include the semantic index). Then we use the expert to provide distillation supervision for joint optimization of student EM modules with MemorySpotter.

For a given input $(\mathcal{V}, \mathcal{Q})$ and ground-truth response $\mathcal{R}$, let $\bar{c}_{\text{expert}}$ and $\bar{c}_{\text{student}}$ be the cross-modal encoder outputs from Equation (4) for the expert and student EM models, respectively. Unlike the student, the expert uses all the video features. The *feature distillation loss* $\mathcal{L}_{\text{FD}}$ trains the student to match the expert's cross-modal features.

$$\mathcal{L}_{\text{FD}} = \left\| \text{StopGrad}(\bar{c}_{\text{expert}}) - \bar{c}_{\text{student}} \right\|_1, \tag{9}$$

where the gradient is not propagated to the frozen expert. Similarly, we also define the *prediction distillation loss* $\mathcal{L}_{\text{PD}}$, which trains the student localization predictions to match the expert localization predictions. For example, in the VSLNet (Zhang et al., 2020a), this would train the student EM model to minimize the KL divergence between its predicted distribution over highlight scores, start and end locations, and the corresponding expert distributions. See Appendix B for more details. Overall, our final loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{EM}} + \lambda_{\text{SEL}}\mathcal{L}_{\text{SEL}} + \lambda_{\text{FD}}\mathcal{L}_{\text{FD}} + \lambda_{\text{PD}}\mathcal{L}_{\text{PD}}, \tag{10}$$

where the λ_* are loss scaling hyperparameters determined via validation. Jointly, they encourage the model to improve EM performance while limiting the budget of clips selected.

4. Experiments

We next validate SpotEM's impact in practice.

4.1. Experimental setup

We evaluate our approach on the large-scale EM NLQ benchmark from Ego4D (Grauman et al., 2022), which is the only public dataset supporting this task to our knowledge. The dataset contains $11.3k/3.9k/4.0k$ queries annotated over $136/45/46$ hours of train/val/test videos. Each video clip is 8.2 minutes on average (with the longest video spanning 20 minutes), and each response window is 10.5 seconds on average. Unlike Ego4D, prior query localization and QA datasets focus only on third-person settings (Regneri et al., 2013; Gao et al., 2017; Tapaswi et al., 2016), or derive videos from simulation with strong assumptions of ground-truth odometry (Datta et al., 2022).

While Ego4D uniquely supports the NLQ task by having long-form egocentric videos and natural language queries, we also test the generality of our approach by benchmarking it on the TACoS dataset for natural language grounding (NLG). Instead of localizing responses to natural language queries in egocentric videos like Ego4D NLQ, the goal in TACoS NLG is to localize short descriptions of the human activities in exocentric videos. It contains long exo videos of kitchen activities (5 minutes on average) and short natural language moments (5 seconds on average).

Evaluation metrics: We measure NLQ and NLG accuracy using MR@1 and MR@5 metrics, which are recall@$\{1, 5\}$ averaged over temporal IoU values $[0.3, 0.5]$. We measure the savings in clip-feature computation using the efficiency-level metric:

$$\eta = \mathbb{E}_{(V,Q)\sim D_{\text{val}}}[100 - k], \tag{11}$$

where $k = 100 \times \frac{\text{\# sampled clips}}{\text{\# total clips}}$. We quantify computational cost with TFLOPs measured using the DeepSpeed library (Rasley et al., 2020). Since the clip feature extraction accounts for over 99.9% of the computational cost, TFLOPs is approximately inversely proportional to η.

4.2. Baselines

We perform experiments with multiple base EM methods and compare against several clip-selection baselines. In particular, we choose three state-of-the-art EM methods to demonstrate the effectiveness and versatility of our clip-selection approach.

InternVideo (Chen et al., 2022) proposes a video foundation model that learns a single video representation to achieve state-of-the-art on several tasks including EM. This method won the ECCV 2022 EM challenge.

ReLER (Liu et al., 2022) uses a modified version of VSLNet-L (Zhang et al., 2021) and proposes video-level data augmentation techniques for NLQ. This method was the winning entry in the CVPR 2022 EM challenge.

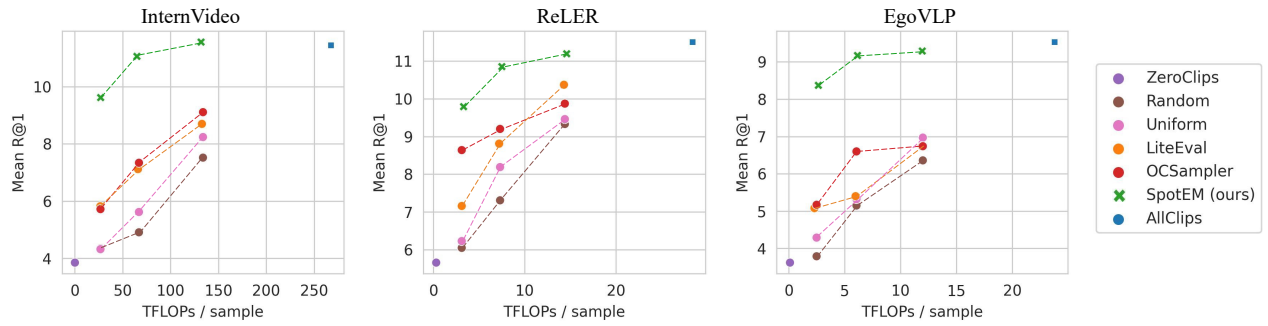

Figure 3. Plot comparing accuracy (y axis) vs. computational cost (x axis) for clip selection methods with InternVideo (Chen et al., 2022), ReLER (Liu et al., 2022), and EgoVLP (Lin et al., 2022b) EM methods. All methods are evaluated on the Ego4D NLQ benchmark. SpotEM achieves over 84% – 97% of the most expensive AllClips method with 4-10× lower computational cost. Note that the computation cost for each clip sampling method includes the cost of extracting semantic index features (see Appendix E for more details). For each method, we show the complete results in Tables 1, 5 and 6.

EgoVLP (Lin et al., 2022b) performs large-scale egocentric video-language pretraining on paired (video clip, narrations text) from Ego4D. This method placed second in the CVPR 2022 EM challenge.

For each base EM method, we compare SpotEM against several clip sampling baselines that represent alternate ways to select clips for reducing the EM inference cost.

ZeroClips: This does not sample any clip features, and only uses the (cheap) semantic indexing features for EM. This serves as a lower bound for clip sampling methods and has the lowest computational cost.

Random: This randomly samples $k\%$ of clips.

Uniform: This samples $k\%$ of clips, uniformly spaced.

LiteEval: This is our implementation of the model of Wu et al. (2019a), which linearly scans the video clips while making binary choices for clip selection. We modified it to include query features as additional inputs to the model.

OCSampler: This is our implementation of the model of Lin et al. (2022a), which previews each video clip independently, predicts per-clip selection scores and selects the top $k\%$ clips with the highest scores. We modified it to include query features as additional inputs.

AllClips: This is the base EM model which performs the task using all video clips (i.e., no clip sampling). This serves as an upper bound for clip sampling methods and has the highest computational cost.

For all baselines other than AllClips, we use ImageNet semantic indexing features in addition to the selected clip features for performing EM. Please see Appendix D for implementation details.

4.3. Experimental results

In Table 1, we compare SpotEM with naïve baselines and prior state-of-the-art clip selection methods (Wu et al., 2019a; Lin et al., 2022a) on the Ego4D NLQ benchmark. The base EM method is the state-of-the-art InternVideo method (Chen et al., 2022). We evaluate methods at different efficiency levels $\eta = [50, 75, 90]$. For random, uniform, and all learned methods, we train one model per efficiency level η. For random, uniform and OCSampler, this means sampling $k\% = 100 - \eta$ of clips during training and evaluation. For the LiteEval and SpotEM, we set $\gamma = 1 - \frac{\eta}{100}$ in Equation (8) during training. Note that the efficiency for LiteEval and SpotEM may be slightly larger than the specified η since the selection is learned end-to-end without any manual intervention.

Consider rows 1-5 in Table 1, which presents a comparison across all methods with ImageNet features as the semantic index. Among the baselines, we observe a consistent trend:

$$\text{OCSampler} \geq \text{LiteEval} > \text{Uniform} \geq \text{Random} > \text{ZeroClips}$$

As expected, all methods are better than the lower-bound zero baseline. Uniform sampling captures information more effectively than random sampling. Importantly, all learned methods outperform naïve baselines by a good margin, confirming the value of intelligent clip selection for EM.

Our SpotEM in row 5 outperforms all the baselines, even before incorporating our RIO semantic index and distillation losses, showing the strength of our MemorySpotter architecture. When compared to the linear clip-wise scan strategy in LiteEval, our method previews the entire video using transformer-based attention and makes selection decisions for all clips at once. This results in 1 – 3 higher absolute mean-recall (MR) than LiteEval. Our method also obtains 0.5 – 2.5 higher absolute MR than OCSampler. Unlike OCSampler, which performs one-shot score prediction for all clips, SpotEM performs recursive clip selection by

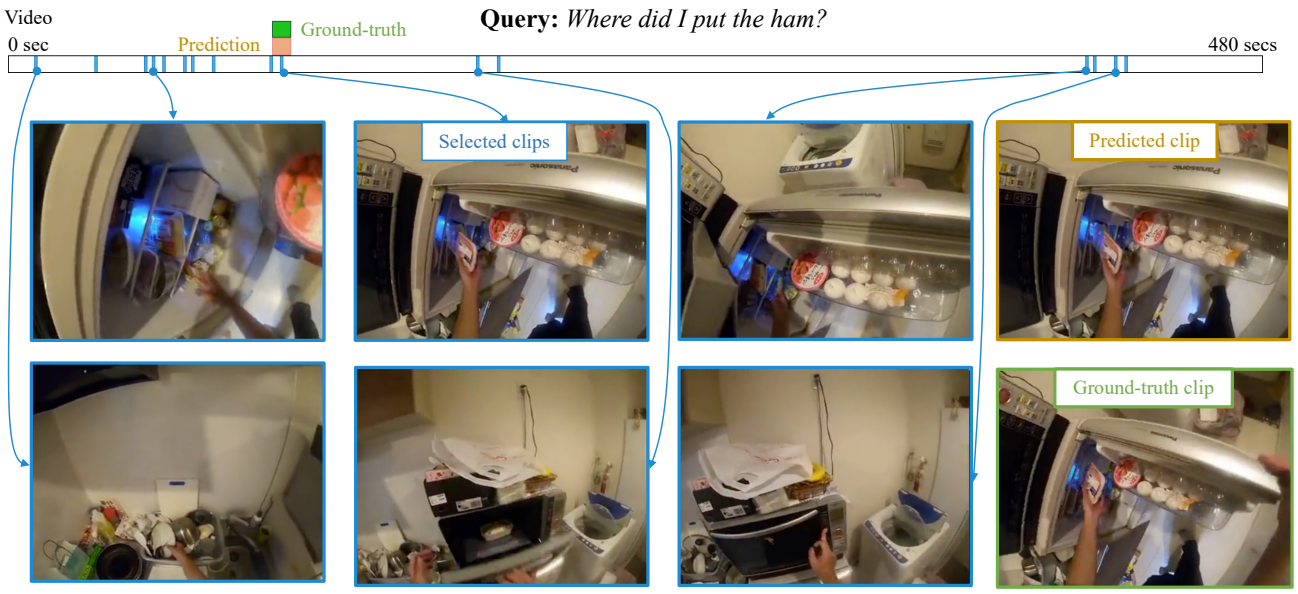

Figure 4. We visualize an example of how SpotEM efficiently performs EM. The query is shown on the top. Below the query, we show a temporal plot containing the video (8 minutes long) overlayed with the timestamps of the clips selected by the policy in light blue (a ∼0.5 seconds chunk of video). The prediction (yellow) and ground truth (green) are shown above the video plot. For a representative set of clips selected by the model, we show the image taken from the center of the clip (highlighted in blue), followed by images from the center of the predicted (yellow) and ground-truth (green) time windows. **SpotEM** uses RIO semantic index to preview the video cheaply, identifies clips that are likely to contain the interaction (put, ham), and correctly localizes the response. In Appendix J, we present more examples that highlight the effective clip selection strategies learned by SpotEM. We also present failure cases where SpotEM confuses closely related objects (e.g., salad dressing vs. salt container) and fine-grained attributes of objects (e.g., brown box vs. blue box).

previewing the video multiple times. At each step, it selects a subset of video clips, extracts their (heavier) clip features, and uses this knowledge for future steps. This results in better performance, especially for higher efficiency levels. This comparison is valuable to show how MemorySpotter's design overcomes limitations of SoTA sampling models designed for action recognition rather than episodic memory.

While the SpotEM clip selection strategy from row 5 outperforms prior methods, it falls short of the original InternVideo results (NB: the latter uses dramatically more computation). However, when we replace the ImageNet features with our RIO features in row 6, we observe good improvements of 0.5 – 2 MR with little to no increase in the computational cost. This confirms the value of building a (cheap) semantic index that captures room, interaction, and object features for efficient EM. When we further add distillation losses to improve model optimization in row 7, the performance improves by 1.5 – 2 MR and effectively bridges the gap between our efficient SpotEM and the base (status quo) EM method which uses all clips. Specifically, we can achieve 84%, 96.5% and 100% of original MR@1 metric with 10×, 4× and 2× reduction in computational cost, respectively. By combining our novel clip selection model, RIO semantic index, and distillation losses during training, SpotEM successfully reduces the computational cost by 4× while maintaining over 95% of the EM performance. Please see Figure 3 (left) for a performance vs. cost chart summarizing our findings. We present a qualitative analysis of SpotEM in Figure 4.

We also test SpotEM on two more EM models: ReLER (Liu et al., 2022) in Figure 3 (center) and EgoVLP (Lin et al., 2022b) in Figure 3 (right). In both cases, SpotEM outperforms all baselines across efficiency levels, and attains at least 95% of the original EM method's accuracy with a 4× reduction in computational cost. See Appendix F for results analogous to Table 1 for ReLER and EgoVLP. The trends are similar as those for InternVideo above, confirming that our approach generalizes across multiple EM methods.

In Appendix K, we compare clip sampling methods on the TACoS dataset for natural language grounding. The trends largely echo our results on the Ego4D NLQ benchmark and confirm the advantages of SpotEM for NLG on exo videos. However, we find that the RIO features are not beneficial, likely due to the ego-exo domain shift since RIO features are trained on egocentric images.

4.4. Ablation studies

Thus far, we compared SpotEM with several clip-selection baselines and demonstrated its superiority. We now present

ablation studies on the Ego4D NLQ benchmark to analyze different aspects of our model.

Ablation of RIO semantic index: In Section 4.3, we confirmed the benefits of using RIO features over standard ImageNet features. We now study the impact of removing each RIO feature, one by one, on the EM task (see Table 2). The base EM method is InternVideo. We study SpotEM without distillation losses to avoid exhaustively training expert models for each feature set. At $\eta = 50\%$, the impact of removing any one feature is minimal since 50% of the expensive clip features are already available. At higher efficiencies, the performance noticeably decreases when we remove any one feature. This study confirms our intuitions: RIO semantic index captures complementary aspects of the EM task and are critical to spotting query-relevant clips.

In Appendix G, we analyze the impact of removing RIO features across query templates (e.g., *where is object X? what did I put in X? what is X before/after event Y?*) and study the role of RIO features for performing EM efficiently. Our study indicates that each of the RIO features has a varying impact on the templates. Removing room features affects queries with strong scene association (e.g., ovens are in kitchens), while removing interaction features affects queries that require object-interaction reasoning (e.g., *"how many drawers did I open?"*). We also found that RIO features facilitate intelligent clip-selection, but do not improve the absolute EM performance themselves.

Ablation of recursive clip selection: Unlike prior methods by Wu et al. (2019a) and Lin et al. (2022a) that were designed to handle shorter videos for recognition, we explicitly target longer videos for EM using our recursive clip-selection approach, where we preview the entire video, and actively select subsets of clips over multiple steps. We assess the impact of the recursion length N in Figure 5. Specifically, we train SpotEM models with $\eta = [50, 75, 90]$ and $N = [1, 2, 4, 8]$, where $N = 4$ is our default choice. $N = 1$ is the non-recursive case where the model looks at only the semantic index for selecting all clips, without incorporating any heavy clip features for selection. Across efficiency levels, we observe that the mean recall@5 increases from $N = 1$ to $N = 4$, confirming the value of recursively selecting clips and incorporating knowledge from prior clips for future selection. The performance reduces beyond $N = 4$ since gradient propagation becomes more challenging.

We further analyze SpotEM's performance as a function of video duration in Appendix H and find that it continues to perform well on long videos, outperforming the baseline sampling methods. SpotEM also achieves 85% of the base EM method's performance, while sampling only 25% of the clips. However, the absolute performance is ultimately limited by the underlying EM method. We also present a detailed study of SpotEM's clip selection behaviors in Appendix I. Our experiments verify our intuition that SpotEM samples query-relevant clips that are useful for the base EM method to respond to the query. SpotEM does not try to respond to the query directly on its own.

			MR@1 at η			MR@5 at η		
R	I	O	50	75	90	50	75	90
✓	✓	✓	9.84	**9.92**	**7.48**	**18.70**	**17.27**	**16.83**
	✓	✓	9.83	9.26	7.26	17.38	17.08	13.36
✓		✓	9.52	8.36	5.89	17.23	14.83	11.05
✓	✓		**10.38**	9.04	6.94	18.53	16.30	13.12

Table 2. **Ablation study of semantic index:** We assess the impact of removing each one of the RIO features on EM performance of InternVideo (Chen et al., 2022). The first 3 columns indicate whether object, room, interaction features are used, respectively. The second row shows the efficiency level η of the model.

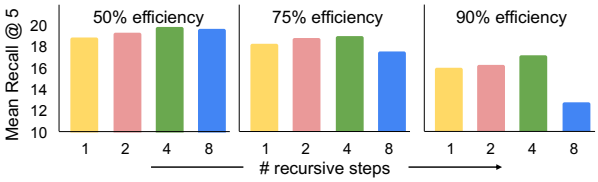

Figure 5. **Ablation study of # recursive steps:** We assess the impact of the recursive step size N for SpotEM on MR@1.

5. Conclusions

SpotEM tackles the efficiency challenge of answering episodic memory queries on long egocentric video head-on. Our novel MemorySpotter clip selection policy, learned jointly with the EM model, leverages cheap video features to prioritize the temporal regions most likely to yield a given natural language query's answer. Together with our novel distillation losses and well-designed RIO semantic indexing features, SpotEM successfully reduces the cost for multiple SoTA EM methods—4 to 10× efficiency gains while maintaining high accuracy. While the videos in our study already represent a sizeable jump in length compared to mainstream video understanding work, in future work, we are interested in exploring a hierarchical variant of our model to triage the video across hours of content.

6. Acknowledgements

UT Austin is supported in part by the IFML NSF AI Institute and NSF CCRI. KG is a paid as a researcher at Meta. We thank the ICML reviewers and area chair for their valuable feedback.

References

Abu Farha, Y., Richard, A., and Gall, J. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5343–5352, 2018.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Bertasius, G., Wang, H., and Torresani, L. Is space-time attention all you need for video understanding? In *ICML*, volume 2, pp. 4, 2021.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970, 2015.

Cai, M., Kitani, K., and Sato, Y. Understanding hand-object manipulation by modeling the contextual relationship between actions, grasp types and object attributes. *arXiv preprint arXiv:1807.08254*, 2018.

Chen, D., Bilgic, M., Getoor, L., and Jacobs, D. Dynamic processing allocation in video. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2174–2187, 2011.

Chen, G., Xing, S., Chen, Z., Wang, Y., Li, K., Li, Y., Liu, Y., Wang, J., Zheng, Y.-D., Huang, B., et al. Internvideo-ego4d: A pack of champion solutions to ego4d challenges. *arXiv preprint arXiv:2211.09529*, 2022.

Chen, Y., Wang, S., Zhang, W., and Huang, Q. Less is more: Picking informative frames for video captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 358–373, 2018.

Damen, D., Leelasawassuk, T., Haines, O., Calway, A., and Mayol-Cuevas, W. W. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, volume 2, pp. 3, 2014.

Damen, D., Doughty, H., Farinella, G. M., , Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130: 33–55, 2022. URL https://doi.org/10.1007/s11263-021-01531-2.

Datta, S., Dharur, S., Cartillier, V., Desai, R., Khanna, M., Batra, D., and Parikh, D. Episodic memory question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19119–19128, 2022.

Del Molino, A. G., Tan, C., Lim, J.-H., and Tan, A.-H. Summarization of egocentric videos: A comprehensive survey. *IEEE Transactions on Human-Machine Systems*, 47(1):65–76, 2016.

Fan, H., Xu, Z., Zhu, L., Yan, C., Ge, J., and Yang, Y. Watching a small portion could be as good as watching all: Towards efficient video classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 705–711. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/98. URL https://doi.org/10.24963/ijcai.2018/98.

Fathi, A., Ren, X., and Rehg, J. M. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pp. 3281–3288, 2011. doi: 10.1109/CVPR.2011.5995444.

Feichtenhofer, C. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 203–213, 2020.

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

Furnari, A. and Farinella, G. M. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4021–4036, 2020.

Furnari, A., Farinella, G. M., and Battiato, S. Recognizing personal locations from egocentric videos. *IEEE Transactions on Human-Machine Systems*, 47(1):6–18, 2016.

Gao, J., Sun, C., Yang, Z., and Nevatia, R. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pp. 5267–5275, 2017.

Gao, R., Oh, T.-H., Grauman, K., and Torresani, L. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Girdhar, R. and Grauman, K. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13505–13515, 2021.

Girdhar, R., Singh, M., Ravi, N., van der Maaten, L., Joulin, A., and Misra, I. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16102–16112, 2022.

Gowda, S. N., Rohrbach, M., and Sevilla-Lara, L. Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1451–1459, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Hazan, T. and Jaakkola, T. S. On the partition function and random maximum a-posteriori perturbations. In *ICML*, 2012.

Hou, Z., Zhong, W., Ji, L., Gao, D., Yan, K., Chan, W.-K., Ngo, C.-W., Shou, Z., and Duan, N. An efficient coarse-to-fine alignment framework@ ego4d natural language queries challenge 2022. *arXiv preprint arXiv:2211.08776*, 2022.

Jang, E., Gu, S., and Poole, B. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Jiang, Y.-G., Dai, Q., Mei, T., Rui, Y., and Chang, S.-F. Super fast event recognition in internet videos. *IEEE Transactions on Multimedia*, 17(8):1174–1186, 2015. doi: 10.1109/TMM.2015.2436813.

Jiang, Y.-G., Wu, Z., Wang, J., Xue, X., and Chang, S.-F. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):352–364, 2018. doi: 10.1109/TPAMI.2017.2670560. URL https://doi.org/10.1109/TPAMI.2017.2670560.

Kazakos, E., Nagrani, A., Zisserman, A., and Damen, D. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5492–5501, 2019.

Korbar, B., Tran, D., and Torresani, L. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6232–6242, 2019.

Lee, Y. J. and Grauman, K. Predicting important objects for egocentric video summarization. *International Journal on Computer Vision*, 2015.

Li, H., Wu, Z., Shrivastava, A., and Davis, L. S. 2d or not 2d? adaptive 3d convolution selection for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6155–6164, 2021.

Lin, J., Duan, H., Chen, K., Lin, D., and Wang, L. Ocsampler: Compressing videos to one clip with single-step sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13894–13903, 2022a.

Lin, K. Q., Wang, A. J., Soldan, M., Wray, M., Yan, R., Xu, E. Z., Gao, D., Tu, R., Zhao, W., Kong, W., et al. Egocentric video-language pretraining. *arXiv preprint arXiv:2206.01670*, 2022b.

Liu, N., Wang, X., Li, X., Yang, Y., and Zhuang, Y. Reler@ zju-alibaba submission to the ego4d natural language queries challenge 2022. *arXiv preprint arXiv:2207.00383*, 2022.

Maddison, C., Mnih, A., and Teh, Y. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

Meng, Y., Lin, C.-C., Panda, R., Sattigeri, P., Karlinsky, L., Oliva, A., Saenko, K., and Feris, R. Ar-net: Adaptive frame resolution for efficient action recognition. In *European Conference on Computer Vision*, pp. 86–104. Springer, 2020.

Mo, S., Mu, F., and Li, Y. A simple transformer-based model for ego4d natural language queries challenge. *arXiv preprint arXiv:2211.08704*, 2022.

Nagarajan, T., Li, Y., Feichtenhofer, C., and Grauman, K. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 163–172, 2020.

Nagarajan, T., Ramakrishnan, S. K., Desai, R., Hillis, J., and Grauman, K. Egocentric scene context for human-centric environment understanding from video. *arXiv preprint arXiv:2207.11365*, 2022.

Ortis, A., Farinella, G. M., D'Amico, V., Addesso, L., Torrisi, G., and Battiato, S. Organizing egocentric videos of daily living activities. *Pattern Recognition*, 72:207–218, 2017.

Panda, R., Chen, C.-F. R., Fan, Q., Sun, X., Saenko, K., Oliva, A., and Feris, R. Adamml: Adaptive multi-modal learning for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7576–7585, October 2021.

Price, W., Vondrick, C., and Damen, D. Unweavenet: Unweaving activity stories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13770–13779, 2022.

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 *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3406703. URL https://doi.org/10.1145/3394486.3406703.

Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., and Pinkal, M. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013.

Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., and Schiele, B. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pp. 184–195. Springer, 2014.

Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., and Schiele, B. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Suin, M. and Rajagopalan, A. An efficient framework for dense video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12039–12046, 2020.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

Tong, Z., Song, Y., Wang, J., and Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022.

Wu, C.-Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., and Krähenbühl, P. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6026–6035, 2018.

Wu, Z., Xiong, C., Jiang, Y.-G., and Davis, L. S. Liteeval: A coarse-to-fine framework for resource efficient video recognition. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/bd853b475d59821e100d3d24303d7747-Paper.pdf.

Wu, Z., Xiong, C., Ma, C.-Y., Socher, R., and Davis, L. S. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1278–1287, 2019b.

Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., and Zhuang, Y. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

Xu, H., Ghosh, G., Huang, P.-Y., Arora, P., Aminzadeh, M., Feichtenhofer, C., Metze, F., and Zettlemoyer, L. Vlm: Task-agnostic video-language model pre-training for video understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4227–4239, 2021.

Yeung, S., Russakovsky, O., Mori, G., and Fei-Fei, L. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2678–2687, 2016.

Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2718–2726, 2016.

Zhang, H., Sun, A., Jing, W., and Zhou, J. T. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6543–6554, 2020a.

Zhang, H., Sun, A., Jing, W., Zhen, L., Zhou, J. T., and Goh, R. S. M. Natural language video localization: A revisit in span-based question answering framework. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

Zhang, S., Peng, H., Fu, J., and Luo, J. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12870–12877, 2020b.

Zhou, Y. and Berg, T. L. Temporal perception and prediction in ego-centric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4498–4506, 2015.

Zhu, L., Tran, D., Sevilla-Lara, L., Yang, Y., Feiszli, M., and Wang, H. Faster recurrent networks for efficient video classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13098–13105, 2020.

Appendix: Table of contents

A. Description of EM methods
B. SpotEM distillation losses
C. Gumbel-softmax for clip sampling
D. Implementation details
E. Calculating computational cost for SpotEM
F. Complete results for EgoVLP, ReLER
G. Ablation study of RIO features
H. NLQ performance vs. video duration
I. Clip-selection behaviors of SpotEM
J. Qualitative analysis of SpotEM
K. Benchmarking SpotEM on TACoS dataset

A. Description of EM methods

We experiment with three different EM methods in the main paper (Liu et al., 2022; Lin et al., 2022b; Chen et al., 2022). Figure 6 depicts the working of an abstracted episodic memory architecture that encapsulates these three methods. We now provide more details about the individual methods.

InternVideo (Chen et al., 2022) proposes a video foundation model that learns a single video representation to achieve state-of-the-art on several tasks including EM. It pretrains two VideoMAE video backbones (Tong et al., 2022) to predict verbs and nouns associated with Ego4D clips, respectively. Overall, it combines the EgoVLP TimeSformer, VideoMAE-verb, and VideoMAE-noun backbones to obtain visual representations. It extracts query-text features using the DistillBERT backbone pretrained by Lin et al. (2022b). It uses the VSLNet architecture for temporal localization (Zhang et al., 2020a), where the cross-modal encoder consists of a transformer encoder to independently encode video and query features, and a context-query attention module to obtain cross-modal embeddings (Seo et al., 2016). A span-based localizer is used to localize the response. It consists of a highlight predictor that predicts the probability of each clip overlapping with the response, and uses normalized probabilities to re-weight the video features for localization. A localizer the predicts the start-end probabilities for the resonse conditioned on the output of the highlight module. Please see Appendix B for more details. This method won the ECCV 2022 EM challenge.

ReLER (Liu et al., 2022) uses a modified version of VSLNet-L (Zhang et al., 2021) and proposes video-level data augmentation techniques for NLQ. It uses a TimeSformer backbone pretrained from (Lin et al., 2022b) and a pretrained CLIP image encoder (Radford et al., 2021) as video encoders. The text backbone is a pretrained CLIP text encoder. It adapts the VSLNet-L architecture for temporal localization (Zhang et al., 2021). The cross-modal encoder consists of a multi-scale transformer encoder to

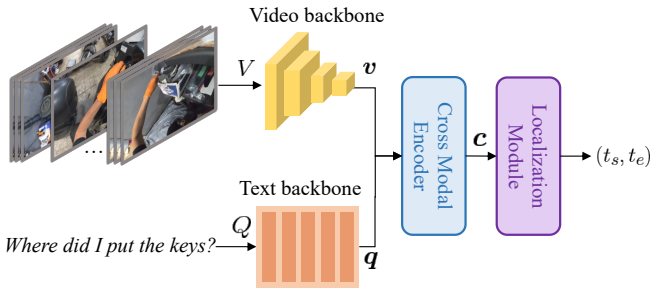

Figure 6. **Episodic Memory architecture:** We depict how an abstracted episodic memory model works for the natural language queries task. Given video V and text query Q, it extracts corresponding features $\boldsymbol{v}$ and $\boldsymbol{q}$ using pretrained backbone models. A cross-modal encoder is used to jointly reason across the two modalities and obtain a cross-modal embedding $\boldsymbol{c}$. The localization model predicts the location of the response conditioned on the cross-modal embedding.

encode video clips, a transformer encoder for the query, and a cross-modal attention mechanism to obtain cross-modal embeddings. A span-based localizer is used as the localization module (similar to VSLNet). Please see (Liu et al., 2022) for more details. This method was the winning entry in the CVPR 2022 EM challenge.

EgoVLP (Lin et al., 2022b) performs large-scale egocentric video-language pretraining on paired (video clip, narrations text) from Ego4D. This method placed second in the CVPR 2022 EM challenge. It pretrains a TimeSformer (Bertasius et al., 2021) video backbone and a DistillBERT (Sanh et al., 2019) text backbone. Similar to InternVideo, EgoVLP uses the VSLNet architecture for temporal localization (Zhang et al., 2020a).

B. SpotEM distillation losses

We now describe the VSLNet architecture (Zhang et al., 2020a), its EM task losses, and how we apply our distillation losses to this model (refer Section 3.5).

VSLNet architecture

We first overview the architecture of VSLNet. Specifically, we use the variant proposed by Grauman et al. (2022), where the recurrent modules in the original model from Zhang et al. (2020a) are replaced by transformer modules. Let $\boldsymbol{v}$ and $\boldsymbol{q}$ denote the video and query features obtained from the respective backbones (see Figure 6). A cross-modal encoder is used to perform cross-modal reasoning:

$$\boldsymbol{c} = \text{CrossModalEncoder}(\boldsymbol{v}, \boldsymbol{q}) \in \mathbb{R}^{L \times D_h}.$$

This consists of a transformer encoder module that independently updates the video features v and query features q by performing self-attention. It then uses the context-query attention mechanism to enhance the video features using information from the query features (Zhang et al., 2020a; Seo et al., 2016).

VSLNet then introduces the query-guided highlighting (QGH) module, which is a 1D convolutional layer that predicts the probability that a clip lies within a temporal neighborhood of the response:

$$\hat{\mathcal{S}}_h = \sigma(\text{Conv1D}(\boldsymbol{c})) \in \mathbb{R}^{L \times 1}, \tag{12}$$

where σ is the sigmoid activation function. These probabilities are used to re-weight the cross-modal features $\boldsymbol{c}$:

$$\boldsymbol{c}_h = \hat{\mathcal{S}}_h \cdot \boldsymbol{c} \in \mathbb{R}^{L \times D_h}.$$

Finally, VSLNet uses a conditioned span predictor to infer the probabilities of each feature location being the start and end points of the response window:

$$\hat{p}_s, \hat{p}_e = \text{ConditionalSpanPredictor}(\boldsymbol{c}_h), \tag{13}$$

where $\hat{p}_s, \hat{p}_e \in \mathbb{R}^{L \times 1}$ are log-probabilities per feature location and "ConditionalSpanPredictor" consists of a transformer encoder for performing self-attention and an MLP to predict the log-probabilities. Next, we describe the loss functions proposed in VSLNet for the EM task.

EM task losses

VSLNet trains the model using a span loss and a QGH loss. The span loss is used to supervise the start/end probability predictions from Equation (13):

$$\mathcal{L}_{\text{span}} = \frac{1}{2}\left[f_{\text{CE}}(\hat{p}_s, p_s^*) + f_{\text{CE}}(\hat{p}_e, p_e^*)\right],$$

where p_s^*, p_e^* are the ground-truth labels for the response start and end times, and f_{CE} is the cross-entropy loss function. The QGH loss is used to supervise the query-guided highlighting prediction from Equation (12):

$$\mathcal{L}_{\text{QGH}} = f_{\text{CE}}(\hat{S}_h, S_h^*),$$

where S_h^* is the ground-truth highlight score that covers an extended temporal window around the ground-truth response $\mathcal{R}$. Please see the work from Zhang et al. (2020a) for more details. The overall EM loss combines the span and QGH losses: $\mathcal{L}_{\text{EM}} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}}$.

Distillation losses

In Section 3.5, we introduced our distillation losses to break the negative feedback loop when jointly training MemorySpotter and the EM model. Specifically, we trained an

expert EM model that performs the task with all the video and semantic index features without any clip sampling (i.e., no MemorySpotter). We then derived supervision from the expert EM model in the form of distillation losses to train our student model. We proposed the feature distillation loss L_{FD} to train the student model to match the expert's cross-modal features (see Equation (9)). We also proposed the prediction distillation loss L_{PD} to encourage the student's localization predictions to match those of the expert.

We now provide more details about L_{PD} in the context of VSLNet. Let us denote the highlight scores, start and end log-probabilities predicted by the expert as S_h^{expert}, p_s^{expert} and p_e^{expert}, respectively [6] (refer to Equations 12 and 13). Let S_h^{student}, p_s^{student}, p_e^{student} be the corresponding predictions from the student model. Then, the prediction distillation loss is defined as follows:

$$\begin{aligned} L_{\text{PD}} = \lambda_{\text{PD}}^{h} D_{\text{KL}}\Big(S_h^{\text{student}} \,||\, S_h^{\text{expert}}\Big) \\ + \lambda_{\text{PD}}^{l} D_{\text{KL}}\Big(\phi(p_s^{\text{student}}) \,||\, \phi(p_s^{\text{expert}})\Big) \\ + \lambda_{\text{PD}}^{l} D_{\text{KL}}\Big(\phi(p_e^{\text{student}}) \,||\, \phi(p_e^{\text{expert}})\Big), \end{aligned} \tag{14}$$

where ϕ is the softmax activation to convert p_s and p_e into probabilities, λ_{PD}^{h} is the loss scaling for highlight scores, λ_{PD}^{l} is the loss scaling for localization predictions, and D_{KL} is the KL divergence between two probability distributions.

C. Gumbel-softmax for clip sampling

As discussed in Section 3.5, MemorySpotter predicts binary selection values in Equation (7) and is therefore non-differentiable for gradient-based optimization. Prior work has tackled this issue using black-box policy optimization, i.e., reinforcement learning (Yeung et al., 2016; Wu et al., 2019b) or the gumbel-softmax trick which uses a softmax relaxation of argmax sampling during training for differentiability (Hazan & Jaakkola, 2012; Maddison et al., 2017; Jang et al., 2017; Wu et al., 2019a). In this work, we adopt the gumbel-softmax formulation since RL has known issues such as high-variance and instability in training, and requires carefully designed reward functions.

We now formally describe how we use gubmel-softmax trick for training our clip-selection model by adapting notations from Jang et al. (2017). Let b_l be a binary variable specifying whether clip l should be sampled given a video-query pair (V, Q) (refer to Equation (7)). Let π^l be the probability of selecting clip l predicted by the SelectionPolicy. We can then use the gumbel-softmax trick to approximately draw samples b_l from the 2-class categorical distribution with probabilities $[\pi_0, \pi_1]$, where $\pi_0 = \pi^l$ is the probability of sampling clip l and $\pi_1 = 1 - \pi^l$ is the probability of ignoring clip l. The gumbel-softmax distribution is obtained as follows:

$$y_i = \frac{\exp\Big((\log(\pi_i) + g_i)/\tau\Big)}{\sum_{j\in\{0,1\}} \exp\Big((\log(\pi_j) + g_j)/\tau\Big)}, \tag{15}$$

where $i \in [0, 1]$ and τ is the softmax temperature. We then use the "straight-through gumbel-softmax estimator" from (Jang et al., 2017) for training.

During the forward pass, discrete samples b_l are drawn through argmax sampling from the gumbel-softmax distribution in Equation (15).

$$\begin{aligned} \bar{b}_l &= \text{one_hot}\big(\text{argmax}(y_0, y_1)\big) \\ b_l &= \bar{b}_l[0] \end{aligned} \tag{16}$$

This is done for each clip $l \in [1, \cdots, L]$ to obtain the clip selections $\bar{\boldsymbol{b}}$ from Equation (7).

During the backward pass, the gumbel-softmax approximation in Equation (15) is used to compute the gradients. Specifically, we use the selection loss from Equation (8) to encourage the model to select a specified budget of clips and penalize under-/over-sampling. The underlying autograd algorithm (pytorch, in our case) maintains the logits from the forward pass (under the hood) for calculating gradients during the backward pass.

D. Implementation details

We implement all experiments in PyTorch. We modify the implementations of base EM methods to incorporate our SpotEM model and loss functions. For InternVideo and EgoVLP, we use the official NLQ repository released by Grauman et al. (2022). For ReLER, we use the code released by Liu et al. (2022). Unlike the former methods, ReLER performs video-level augmentation, where clips from other videos are randomly concatenated on either sides of a given video. We found it unsuitable to train clip sampling methods like SpotEM and LiteEval using this augmentation since the clip selection budget used in the loss function from Equation (8) is inconsistent between training and inference. For example, if the model is trained to select 10% of clips during training, it may pick more than 10% of clips during inference since there are several irrelevant clips appended to the video during training. Therefore, we pre-train the EM modules for SpotEM and LiteEval using uniform clip sampling + video-level augmentation first, and then learn the clip sampling policies in a second stage of training without video-level augmentation. This was not needed for OCSampler since it deterministically picks the top-k clips during inference (instead of deciding the number of clips to select based on a loss function). We provide the hyperparameters for training SpotEM in Table 3.

[6] The ˆ symbol is ignored for brevity.

	InternVideo / EgoVLP	ReLER
Optimizer	AdamW	AdamW
Encoder hidden size	128	256
# recursive steps (N)	4	4
# training epochs	200	200
Batch size	128	128
Learning rate scheduler	Linear Warmup	-
Initial learning rate	0.001	0.0004
λ_{SEL}	[300.0, 1000.0]	[30.0, 100.0, 300.0]
λ_{FD}	1.0	3.0
λ_{PD}	1.0	1.0
λ^{l}_{PD}	1.0	1.0
λ^{h}_{PD}	[10.0, 30.0]	10.0

Table 3. **Hyperparameters for training SpotEM.** $\lambda_{\text{SEL}}, \lambda_{\text{FD}}, \lambda_{\text{PD}}$ are loss scaling hyperparameters from Equation (10). $\lambda^{l}_{\text{PD}}, \lambda^{h}_{\text{PD}}$ are loss scaling hyperparameters from Equation (14). For hyperparameters with multiple values, we perform a grid-search over the specified values and pick the best model based on validation performance.

Base EM method	Computational cost (in GFLOPs)		
	$\mathcal{C}_v$ (per clip)	$\mathcal{C}_s$ (per image)	$\mathcal{C}_{cs} + \mathcal{C}_{em}$ (per $(\mathcal{V}, \mathcal{Q})$)
EgoVLP	185.8	2.3	7.27
ReLER	220.9	2.3	215.5
InternVideo	2090.8	2.3	7.27

Table 4. Computational cost breakdown for SpotEM

E. Calculating computational cost for SpotEM

In Tables 1, 5 and 6 and Figure 3, we reported the computational cost in terms of averaged TFLOPs per (video, query) pairs. Here, we present a complete breakdown of the compute costs during inference for SpotEM. The overall cost $\mathcal{C}$ for a video-query pair $(\mathcal{V}, \mathcal{Q})$ is calculated as follows:

$$\mathcal{C} = \mathcal{C}_v \times N + \mathcal{C}_s \times L + \mathcal{C}_{cs} + \mathcal{C}_{em}, \tag{17}$$

where N is the number of clips selected by the SpotEM, and L is the total number of video clips. The individual cost terms are described as follows:

$\mathcal{C}_v$ = extract video features per clip (Equation 3)
$\mathcal{C}_s$ = extract semantic-index per image (Equation 1)
$\mathcal{C}_{cs}$ = recursive clip selection for $(\mathcal{V}, \mathcal{Q})$ (Equations 6, 7)
$\mathcal{C}_{em}$ = EM inference for $(\mathcal{V}, \mathcal{Q})$ (Equations 2, 4, 5)

InternVideo and EgoVLP use the VSLNet model with maximum clip length $L = 128$. ReLER uses the VSLNet-L model with maximum clip length $L = 600$. The cost values for SpotEM when integrated with different base EM methods are shown in Table 4.

F. Complete results for EgoVLP, ReLER

Analogous to Table 1 from the main paper, we present results on EgoVLP and ReLER methods in Tables 5 and 6.

Row	Clip selection method	η	Sem. index	MR@1	MR@5	TFLOPs
	ZeroClips	100	ImageNet	3.62	8.19	0.1
1	Random	90	ImageNet	3.79	8.23	2.4
2	Uniform	90	ImageNet	4.29	9.02	2.4
3	LiteEval (Wu et al., 2019a)	90	ImageNet	5.08	10.13	2.2
4	OCSampler (Lin et al., 2022a)	90	ImageNet	5.17	10.62	2.4
5	SpotEM w/o distill (ours)	91	ImageNet	6.00	12.04	2.1
6	SpotEM w/o distill (ours)	90	RIO	7.08	13.73	2.6
7	SpotEM (ours)	90	RIO	**8.37**	**15.26**	2.6
1	Random	75	ImageNet	5.15	10.75	6.0
2	Uniform	75	ImageNet	5.32	10.23	6.0
3	LiteEval (Wu et al., 2019a)	75	ImageNet	5.40	11.56	5.9
4	OCSampler (Lin et al., 2022a)	75	ImageNet	6.60	12.17	6.0
5	SpotEM w/o dilstill (ours)	77	ImageNet	7.34	13.53	5.5
6	SpotEM w/o distill (ours)	76	RIO	8.15	15.59	5.9
7	SpotEM (ours)	75	RIO	**9.17**	**16.44**	6.1
1	Random	50	ImageNet	6.36	12.07	11.9
2	Uniform	50	ImageNet	6.97	13.67	11.9
3	LiteEval (Wu et al., 2019a)	50	ImageNet	6.74	13.27	11.9
4	OCSampler (Lin et al., 2022a)	50	ImageNet	6.74	12.68	11.9
5	SpotEM w/o distill (ours)	51	ImageNet	8.36	15.68	11.8
6	SpotEM w/o distill (ours)	52	RIO	8.44	15.89	11.8
7	SpotEM (ours)	51	RIO	**9.29**	**17.08**	11.9
	AllClips (Lin et al., 2022b)	0	-	9.53	17.50	23.7

Table 5. Comparing efficient clip selection methods for the EgoVLP NLQ method (Lin et al., 2022b) on the Ego4D NLQ benchmark.

We observe trends that are similar to InternVideo, with the exception of SpotEM w/o distill underperforming with ImageNet features for ReLER.

G. Ablation study of RIO features

In Table 2, we studied the effect of RIO features on the overall task performance. We observed that the each of the features were important for obtaining our best performance, especially at higher efficiency levels. We now perform a more detailed study about how these features impact different query types. The NLQ dataset from Grauman et al. (2022) was constructed based on 13 templates spanning questions about objects, places and people. We evaluate the performance of our SpotEM method and the impact of removing different features on each template in Table 7. Since the features are most impactful at higher efficiency levels (as noted in Table 2), we directly evaluate with $\eta = 90$. We only select 10/13 templates that have atleast 100 queries in the validation set. Each of the RIO features has a varying impact on the templates. Removing room features affects 3/10 templates where the query object has strong scene association (e.g., ovens are in kitchens). Removing interaction queries affects 7/10 templates which require reasoning about object-interactions (e.g., *"how many drawers did I open?"*). Finally, removing object features affects 2/10 of queries that require object-oriented reasoning (e.g., *"how many funnels are on the shelf?"*).

We further study the role of RIO features for efficiently performing EM. Specifically, we are interested in knowing if the features only help SpotEM select clips intelligently, or do they help improve the base EM model's performance

Row	Clip selection method	η	LW feats.	MR@1	MR@5	TFLOPs
	ZeroClips	100	ImageNet	5.66	7.52	0.3
1	Random	90	ImageNet	6.05	7.89	3.1
2	Uniform	90	ImageNet	6.23	7.95	3.1
3	LiteEval (Wu et al., 2019a)	90	ImageNet	7.16	9.10	3.1
4	OCSampler (Lin et al., 2022a)	90	ImageNet	8.64	11.14	3.1
5	SpotEM w/o distill (ours)	90	ImageNet	8.19	10.60	2.9
6	SpotEM w/o distill (ours)	90	ORInt	9.58	12.05	3.3
7	SpotEM (ours)	90	ORInt	**9.79**	**12.16**	3.3
1	Random	75	ImageNet	7.31	9.36	7.3
2	Uniform	75	ImageNet	8.19	10.59	7.3
3	LiteEval (Wu et al., 2019a)	75	ImageNet	8.81	11.50	7.2
4	OCSampler (Lin et al., 2022a)	75	ImageNet	9.20	11.65	7.3
5	SpotEM w/o distill (ours)	77	ImageNet	9.59	12.32	6.9
6	SpotEM w/o distill (ours)	75	ORInt	10.33	12.70	7.5
7	SpotEM (ours)	75	ORInt	**10.85**	**13.89**	7.5
1	Random	50	ImageNet	9.33	11.65	14.4
2	Uniform	50	ImageNet	9.46	12.46	14.4
3	LiteEval (Wu et al., 2019a)	50	ImageNet	10.37	13.06	14.3
4	OCSampler (Lin et al., 2022a)	50	ImageNet	9.87	12.41	14.4
5	SpotEM w/o distill (ours)	51	ImageNet	10.18	12.67	14.1
6	SpotEM w/o distill (ours)	50	ORInt	10.58	13.49	14.5
7	SpotEM (ours)	50	ORInt	**11.19**	**14.16**	14.6
	AllClips (Liu et al., 2022)	0	-	11.50	14.65	28.4

Table 6. Comparing efficient clip selection methods for the ReLER NLQ method (Liu et al., 2022) on the Ego4D NLQ benchmark.

as well. To study this, we trained each EM method — InternVideo, ReLER, EgoVLP — by appending the RIO features to the original clip features, and used all clips (i.e., no clip sampling). The results are shown in Table 8. Across methods, we find that RIO features are not helpful when simply concatenated to the original clip features. Their primary utility is to provide cues for intelligent clip selection to responding to a query.

H. NLQ performance vs. video duration

We now study the performance of clip selection methods as a function of the video duration. Longer videos are more challenging since the search space grows significantly and more computational cost may be required for inference. We group the NLQ validation video clips into four buckets: 0-5 mins, 5-10 mikns, 10-15 mins, and 15-20 mins. The statistics of the number of queries and clips in each bucket are shown in Table 9. Majority of the clips are 5-10 mins long. To get reliable statistics for evaluation (i.e., at least 100 queries), we only evaluate on clips in the 5-10 mins and 15-20 mins buckets. We compare the "mean R@1" performance on 5-10 mins clips and 15-20 mins clips for the following methods: AllClips (the upper bound baseline), LiteEval, OCSampler, SpotEM. The base EM method is InternVideo. The results are in Table 10.

We make a few observations. The performance of the AllClips "upper bound" is significantly worse on 15-20 mins clips when compared to 5-10 mins clips, pointing to the difficulty of the task. When sampling only 25% of the clips, SpotEM recovers $\sim 85\%$ of AllClips' performance for 15 - 20 mins clips. While this reduces from $\sim 96\%$ for 5 - 10 mins clips, it is still a relatively high number. Thus, SpotEM continues to work well for longer videos, but the overall performance is ultimately bottlenecked by the underlying EM method. Finally, SpotEM outperforms the learned baselines LiteEval and OCSampler on all cases, confirming its advantages over the baselines.

I. Clip-selection behaviors of SpotEM

As we motivate in Section 1, the key idea behind SpotEM is that not all parts of the video are useful for a given query and there are high-level visual semantics that could steer our attention towards where to look (refer Section 1). Based on this intuition, we decomposed the EM task into two steps: (1) identify query-relevant clips using a high-level preview — the role of SpotEM, and (2) solve the EM task using the selected clips — the role of the base EM model. Importantly, SpotEM is *not* trying to answer the query. Instead, it aims to narrow the search to some relevant clips (and reject several irrelevant clips) and enable the EM model to perform the task using a lower computational budget.

Another reasonable alternative is to directly train an EM model to predict the ground-truth response window using only the semantic index, and use these predicted response windows to select clips for the heavier video feature extraction. Based on this, we created the *direct-supervision* clip selector (as opposed to SpotEM, where we indirectly supervise clip selection based on the EM performance using the selected clips). It works as follows: (1) train a ZeroClips EM model to predict the GT response using RIO features, (2) given an NLQ query during inference, use the ZeroClips model to infer the top-k responses, and (3) select clips that overlap with the top-k responses for expensive feature computation. We then train new EM modules (i.e., CrossModalEncoder and LocalizationModule) to perform NLQ using the clips selected by direct-supervision (i.e., RIO features for all clips + video features for selected clips).

The results are in Table 11. When compared to SpotEM, direct-supervision performs poorly. The problem with direct-supervision is that the semantic indexes we use (RIO, in this case) are low resolution in terms of the semantics encoded and may not be enough to infer the GT response accurately. Given a clip with 16 frames (at 30 FPS), the semantic index only encodes a single image within that clip using an efficient image backbone (refer to Section 3.2). Therefore, methods that directly predict the GT response using the semantic index perform poorly (i.e., direct-supervision), which eventually translates to poor clip selection. This suggests that selecting query-relevant clips (and more importantly, rejecting irrelevant clips) is critical for efficiently performing EM.

To understand the clip-selection behaviors resulting from

R	I	O	Object queries								Place queries	People queries
			Where is X before/after Y?	Where is X?	What did I put in X?	How many X's?	In what location did I see X?	What X did I Y?	What X is Y?	State?	Where did I put X	Who did I interact with during Y?
✓	✓	✓	16.55	9.30	18.89	23.68	8.18	17.72	11.52	16.87	11.09	13.04
	✓	✓	14.77	10.39	16.94	**21.05**	7.43	**13.91**	**8.98**	19.63	11.27	13.04
✓		✓	**11.88**	8.21	**13.02**	**15.79**	**5.76**	**9.77**	**9.18**	18.40	10.64	**9.24**
✓	✓		15.79	8.70	16.94	**19.47**	7.62	**11.42**	11.33	18.10	9.66	13.04

Table 7. **Impact of semantic index on different types of queries**: We split the performance of different feature types reported in Table 2 across query types (shown in row 2). We report the mean recall@5 at the highest efficiency level $\eta = 90$, since the features have the biggest impact at this level. Columns 1-3 indicate the presence of room (R), interaction (I), and object (O) features. Row 3 is our proposed method that uses all features. For rows 4-6 are ablations where one of R/I/O features are removed. We highlight the cases where performance drops by more than 2 points on mean recall (in **red**).

Method	MR@1	MR@5
EgoVLP	**9.54**	**17.50**
EgoVLP + RIO	9.33	16.45
ReLER	**11.50**	**14.65**
ReLER + RIO	10.68	13.10
InternVideo	**11.45**	**20.56**
InternVideo + RIO	10.91	18.94

Table 8. **Role of RIO features for EM:** RIO features are not beneficial when simply concatenated with the base EM model's clip features. Their primary role is to help SpotEM intelligently select clips relevant to the query.

	0-5 mins	5-10 mins	10-15 mins	15-20 mins
# video clips	13	289	0	18
# queries	43	3142	0	344

Table 9. Distribution of queries and video clips as a function of video duration.

	5-10 mins clips				15 - 20 mins clips			
Efficiency	0	50	75	90	0	50	75	90
AllClips	12.22	-	-	-	4.80	-	-	-
LiteEval	-	9.34	7.72	6.29	-	2.03	1.31	1.02
OCSampler	-	9.79	7.72	6.02	-	2.62	3.34	2.47
SpotEM	-	**12.27**	**11.73**	**10.20**	-	**4.65**	**4.07**	**3.20**
% results	-	100.4	95.9	83.4	-	96.8	84.7	66.6

Table 10. **NLQ accuracy vs. video duration.** The last row shows the % of AllClips' results achieved by SpotEM.

Method	Efficiency	MR@1	MR@5
Direct-supervision (top k=1)	90	5.27	9.50
SpotEM	90	**7.48**	**14.82**
Direct-supervision (top k=2)	75	5.48	10.24
SpotEM	75	**9.92**	**17.27**
Direct-supervision (top k=5)	50	7.64	13.23
SpotEM	50	**9.84**	**18.70**

Table 11. EM accuracy comparison between direct-supervision and SpotEM. **Note:** We train both methods without distillation losses for apples-to-apples comparison. For direct-supervision, we train models for $k = 1, 2, 3, 4, 5$ and select the largest k that satisfies the given efficiency budget.

indirectly supervising the clip selection in SpotEM, we empirically study the relationship between the clips selected by SpotEM and the ground-truth response. For SpotEM trained at 90% efficiency, we divide the validation queries into cases where the model gets the prediction right (i.e., MR@5 = 1) and the model gets the prediction wrong (i.e., MR@5 $<$ 1). We then measured two statistics for each case.

(1) mean IoU measures the intersection over union between the clips selected by SpotEM and clips belonging to the GT response, averaged over all queries.
(2) mean nonzero intersection measures the percentage of queries where SpotEM selects at least one clip belonging to the GT response.

Results are shown in Table 12. We make two observations. The mean IoU is low for both correct and wrong cases, i.e., SpotEM does not limit itself to selecting clips within the ground-truth response. The mean nonzero intersection is high for correct cases, i.e., 90% of the time, SpotEM picks *at least one clip* within the GT response when its prediction is right. These results confirm our intuition that SpotEM does not try to answer the query (since the mean IoU is low). Instead, it rejects irrelevant clips and selects relevant clips (including clips within the GT response).

J. Qualitative analysis of SpotEM

Analogous to Figure 4 in the main paper, we visualize five success and two failure cases of SpotEM in the form of videos.[7] In each video, we visualize the clips selected by SpotEM at each step, highlight their relevance to the query (if any), and provide a textual description of SpotEM's behavior. Finally, we visualize the predicted and ground-truth responses. Additionally, we describe why SpotEM performs

[7]Qualitative visualizations are available here: https://utexas.box.com/s/p8iheclayaoth2ey95m8o0w9m97snjmb

SpotEM prediction	mean IoU ($\times 100$)	mean non-zero intersection
right (717/3529 queries)	13.96	90.10
wrong (2812/3529 queries)	3.12	32.79

Table 12. Studying the clip-selection behaviors of SpotEM

poorly in failure cases. In success cases, SpotEM is able to identify query-relevant clips and use them to respond to the query accurately. In failure cases, SpotEM tends to confuse one object for another (e.g., dressing vs. salt container) or confuse object colors (e.g., brown box vs. blue box). These failures could be attributed to the similarity of these objects in the RIO feature space.

K. Benchmarking SpotEM on TACoS dataset

Ego4D, to the best of our knowledge, is the first dataset to introduce the EM task with unique properties of long-form egocentric videos and natural-language query annotations. To test our approach on another dataset, we identified a third-person video dataset that may support the natural-language grounding task with long video and short responses. The TACoS dataset contains long third-person kitchen videos (5 mins on average) with relatively short natural language moments (5 secs) (Rohrbach et al., 2014). Since TACoS has third-person videos, we use clip features from a Slow-Fast model pre-trained on Kinetics 400 (Feichtenhofer et al., 2019). Our base method for TACoS NLG is a VSLNet model with SlowFast features. We compare SpotEM with all baselines across different efficiency levels in Table 13. The trends echo results from Section 4.3 on Ego4D NLQ. There are two key differences. LiteEval performs better than OCSampler (row 3 vs. 4), and adding RIO features is not helpful (row 5 vs. 6), likely due to the ego-exo domain shift (RIO features are trained on egocentric images). Overall, SpotEM outperforms the baselines (row 5) and adding distillation losses improves performance by a good margin (row 6 vs. 7). These results confirm the benefits of SpotEM for natural-language grounding on long exocentric videos.

Row	Clip selection method	η	Sem. index	MR@1	MR@5
	ZeroClips	100	ImageNet	7.52	14.03
1	Random	90	ImageNet	8.09	14.42
2	Uniform	90	ImageNet	8.07	13.79
3	LiteEval (Wu et al., 2019a)	90	ImageNet	12.18	18.10
4	OCSampler (Lin et al., 2022a)	90	ImageNet	10.54	17.46
5	SpotEM w/o distill	91	ImageNet	13.34	19.82
6	SpotEM w/o distill	90	RIO	13.66	19.95
7	SpotEM (ours)	90	RIO	**15.08**	**21.57**
1	Random	75	ImageNet	9.31	17.03
2	Uniform	75	ImageNet	11.00	18.68
3	LiteEval (Wu et al., 2019a)	75	ImageNet	14.66	21.64
4	OCSampler (Lin et al., 2022a)	75	ImageNet	12.40	18.90
5	SpotEM w/o distill	76	ImageNet	15.25	23.60
6	SpotEM w/o distill	75	RIO	15.34	23.61
7	SpotEM (ours)	76	RIO	**17.12**	**24.98**
1	Random	50	ImageNet	12.96	21.00
2	Uniform	50	ImageNet	14.66	23.97
3	LiteEval (Wu et al., 2019a)	50	ImageNet	15.56	23.44
4	OCSampler (Lin et al., 2022a)	50	ImageNet	13.56	20.46
5	SpotEM w/o distill	53	ImageNet	16.44	24.81
6	SpotEM w/o distill	52	RIO	16.71	25.07
7	SpotEM (ours)	51	RIO	**17.90**	**26.84**
	AllClips (Grauman et al., 2022)	0	-	18.27	26.54

Table 13. Comparing efficient clip selection methods on TACoS NLG for the VSLNet baseline with SlowFast features (Feichtenhofer et al., 2019).