Episodic Memory Verbalization using Hierarchical Representations of Life-Long Robot Experience

Leonard Bärmann[1], Chad DeChant[2], Joana Plewnia[1], Fabian Peller-Konrad[1],
Daniel Bauer[2], Tamim Asfour[1] and Alex Waibel[1]

Abstract— Verbalization of robot experience, i. e., summarization of and question answering about a robot's past, is a crucial ability for improving human-robot interaction. Previous works applied rule-based systems or fine-tuned deep models to verbalize short (several-minute-long) streams of episodic data, limiting generalization and transferability. In our work, we apply large pretrained models to tackle this task with zero or few examples, and specifically focus on verbalizing life-long experiences. For this, we derive a tree-like data structure from episodic memory (EM), with lower levels representing raw perception and proprioception data, and higher levels abstracting events to natural language concepts. Given such a hierarchical representation built from the experience stream, we apply a large language model as an agent to interactively search the EM given a user's query, dynamically expanding (initially collapsed) tree nodes to find the relevant information. The approach keeps computational costs low even when scaling to months of robot experience data. We evaluate our method on simulated household robot data, human egocentric videos, and real-world robot recordings, demonstrating its flexibility and scalability.

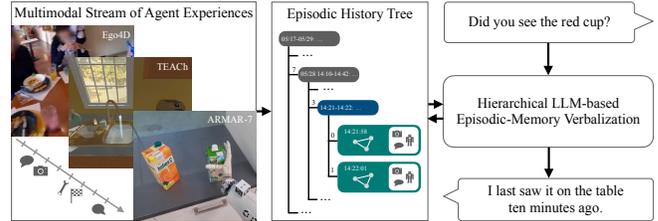**Code, data and demo videos at hierarchical-emv.github.io.**

Fig. 1. Our system answers queries about life-long experience of an agent (human or robotic) by exploring a tree representation of episodic memory.

I. INTRODUCTION

Verbalizing their own experiences is an important ability robots should have to improve natural and intuitive human-robot interaction [1], [2], [3], [4]. It involves summarization of and question answering (QA) about a robot's past actions, observations and interactions, such as the dialog shown on the right of Fig. 1. Building a representation of an agent's Episodic Memory (EM) [5] is crucial to enable such verbalizations, as a system must efficiently store the information from the continuous stream of experience, organize it, and retrieve relevant past events from its EM in response to a user's query. This is particularly challenging as the time horizon of the EM grows.

Existing work on Episodic Memory Verbalization (EMV) either relies on rule-based verbalization of log files [2], [3], or fine-tuning deep models on hand-crafted or auto-generated datasets [1], [6] to perform QA and summarization tasks given the recorded experiences. Both approaches are limited, as they require designing vast numbers of rules or collecting large amounts of experience data.

[1]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany baermann@kit.edu, asfour@kit.edu
[2]Computer Science Department, Columbia University, NY, United States chad.dechant@columbia.edu

To avoid training a system, which typically entails collecting large amounts of multimodal experience data, previous works [7], [8] use language-based representations of the past which can be obtained from pretrained multimodal models. Given such a language-based representation of an agent's history of episodic events, a straightforward way to perform QA is to pass the question and the history to a large language model (LLM), and prompt it to produce an answer. While this works nicely for short histories [7], [9], in this paper, we focus on how to scale such approaches for verbalization of *life-long* experience streams. Although recent LLMs offer increasingly long context windows (i. e., the maximum number of tokens they can process), up to 2M tokens [10], previous studies [11], [12] have shown that these models have difficulty in using all information contained in such long contexts. Furthermore, the computation of transformer models scales quadratically with context length – reducing the number of tokens is thus time- and cost-effective.

Therefore, to scale EMV to life-long experience streams while maintaining a low token budget, we propose to derive a tree-like representation from EM and use an LLM agent for QA to interactively search the tree to find relevant information. Our system, H-EMV (Hierarchical Episodic Memory Verbalization, Fig. 1), processes the continuous stream of experiences and inserts it into a hierarchical representation of the robot's history of episodic events. Different levels of this hierarchy represent different abstraction levels, with the lowest level being raw observations and proprioception and higher levels being represented as natural language concepts. An LLM is prompted for segmentation and summarization in order to recursively create higher-level abstractions. To process queries to the EM, we repurpose the interactive prompting scheme described in our previous work [13]. An LLM is provided with the user's query and functions to access the history tree, and the LLM itself decides which

functions to use in order to fulfill the query, i.e., to answer the question or provide a summary. Since the history tree will grow large over time, we apply an interactive semantic search, inspired by work on robot navigation [14]. Specifically, we contract the tree, i.e., the LLM first sees only the top-level node, and then interactively explores the graph to retrieve the information relevant to the query.

To evaluate our system, we use simulated household episodes from TEACh [15], real-world egocentric human recordings from Ego4D [16], and real-world robot episodes from ARMAR-7, the newest member of the ARMAR humanoid robot family developed at KIT [17]. Our experiments show that H-EMV efficiently scales to extremely long histories of multiple simulated months or real-world human egocentric videos of over six hours, outperforming several baselines and ablations. Real-world robot demonstrations showcase the applicability of our system. We provide our code, evaluation data, and demonstration videos at hierarchical-emv.github.io.

II. RELATED WORK

Episodic Memory for Robots: The concept of EM stems from human cognition [5] and is useful for various technologies including smart wearables [16], [18], smart rooms [19], and especially robotics. For instance, robotic EM can be represented using latent vectors created by deep neural models [20], [1], [21], or by explicitly storing relevant information in a memory system [22], [23], [24]. Another approach is to represent the history of episodic events as text produced by pretrained models [7], [8]. In this paper, we also represent the history tree in form of text, following REFLECT [8] for the broad structure of the hierarchy's lower levels. However, we extend this by adding hierarchical summarization. Furthermore, our multimodal episodic history tree can be dynamically explored by an LLM to gather information from all levels, including the raw observations.

Robot Experience Verbalization: The first work to introduce the term of "verbalizing" robot experiences was [2]. With a rule-based system, they converted a navigation route taken by a mobile service robot to natural language. [3] adapted this framework to verbalization of manipulation activities performed by a humanoid household robot. Similarly, [25] use templates to convert their robot's observations and actions to natural language. More recent works phrase EMV in a more interactive setting, defined as summarization and QA on robot experiences [4]. Both [1], [6] propose end-to-end trained networks receiving multimodal experiences and a question to produce an answer. While [6] work on visual data only, [1] additionally use symbolic and subsymbolic information from the robot's task execution and perception components. Both train on data from simulated household tasks. In contrast to these systems, H-EMV uses pretrained foundation models and does not require additional training data, thus increasing its versatility and easing deployment to the real world. Similar to our setting, QA from streaming data [26], [27] tackles the problem of answering questions based on a long stream of data, where the question is not known in advance and the raw data cannot be stored. However, we apply this to robotics, and approach it with an interpretable, modular system, instead of end-to-end trained memory models.

Video Understanding: Video Understanding, especially Video Question Answering (VideoQA), is related to EMV as it also involves QA on a data stream, which, however, is only a video instead of a multimodal robotic experience stream. VideoQA is an active research area [28] where current major challenges include long-form videos beyond clips of a few seconds as well as egocentric video understanding. Ego4D [16] is a large collection of unconstrained egocentric videos showing daily activities of human camera wearers. Ego4D GoalStep [29] and HCap [9] provide hierarchical annotations for subsets of Ego4D, facilitating reasoning on different abstraction levels. Recent long-form egocentric VideoQA benchmarks include QAEGO4D [18] and EgoSchema [30].

Recent methods for VideoQA can be grouped into (i) end-to-end approaches [31], [32], [33], [9], [34] that typically connect pretrained frozen visual encoders with LLMs by some trained adapter, and (ii) training-free "socratic" [7] approaches [35], [36], [37], [38], [39], [40], [41] that invoke various off-the-shelf models to convert the video into text to be processed by a few-/zero-shot prompted LLM. For instance, [39], [41] use video captioning to produce a transcript of the video and then apply an LLM for QA based on this transcript. VideoTree [36] adaptively selects the frames to caption using a top-down query-relevance-based tree expansion instead of uniform sampling. [40] generate executable Python code from a question, invoking different APIs to query visual and language foundation models. MoReVQA [38] decomposes this into multiple stages, making the LLM's job easier at each stage by focusing on either event parsing, grounding, or reasoning, instead of all at once. In contrast to these predefined prompting schemes, both [35], [37] use an LLM as an agent to analyze the video content in an interactive loop. While [37] iteratively ask the LLM whether to gather more detailed information (by captioning more intermediate frames) or produce the final answer, [35] provide the LLM with API functions invoking tools to search in a database of tracked objects or a memory of frame captions.

Our method similarly treats the LLM as an agent, thus not relying on any predefined information flow. However, we use the full flexibility of code [42] instead of single API calls like in [35], [37]. Compared to VideoTree [36], our history tree is constructed independently of the user's query, since future questions cannot be known in advance in realistic settings, and storing lifelong "raw" video experiences is prohibitive [18]. In contrast to all of the above works, we consider real-world dates and times an integral part of the process. While the recent work TimeChat [43] is also time-sensitive, they refer to video timestamps instead of real-world date-times. Furthermore, and most crucially, we deal with long sequences of multimodal *robotic* experiences, with the longest experiment having over six hours of video or nearly two months on a simulated timeline.

III. METHOD

Our goal is to enable an artificial agent to verbalize and answer questions about its past. Given the continuous, multimodal stream of experiences of a robot agent, we build up a hierarchical and interpretable representation of EM (Sec. III-A). When a user later asks a question, an LLM interactively explores the history tree to gather relevant information, detailed in Sec. III-B.

A. Episodic Memory Construction

From a stream of multimodal robot experiences, we derive a hierarchical representation of the robot's EM, a *history tree*, as shown in Fig. 2, with the lower levels broadly following [8]. Specifically, the tree's levels are:

L0 – Raw Experiences: Leaf nodes collect the raw information available at a specific timestep during the robot's task execution. This includes all modalities that can be perceived by the robot: RGB and depth camera images and recorded audio, as well as information deduced from this data, i.e., recognized objects, their positions, and a text transcription of the audio, if there is user speech. Furthermore, we include everything the agent knows about its state: robot proprioception (joint configuration, mobile platform position), symbolic information about the current action and goal, and text to be spoken by the robot's text-to-speech component.

L1 – Scene Graphs: The first level of non-leaf nodes in the history tree has a one-to-one mapping to the L0 leafs. On this level, we derive a scene graph from the given observations, consisting of the detected objects as nodes and their spatial relations (e.g., on top, inside) as edges. The exact method for constructing the scene graph varies in our experiments. For the pure vision-based approach, objects are detected using pretrained models and heuristics are applied to infer semantically meaningful relations [8]. In our real-robot experiments, we use the existing components in our robot software framework ArmarX [44] that already provide semantic scene information.

L2 – Events: Next, we group and summarize the nodes from the previous level based on changes in the scene graph, the currently executed action or goal, as well as when there is a new speech recognition. We also create a template-based natural-language summary, including the latest scene graph, the current action, and recognized speech command. In our real-robot experiments, we use an LLM to filter and summarize the raw action parameters, which would be excessively detailed otherwise.

L3 – Goals: Based on the current goal from the L0 node, we group event nodes and again create a rule-based natural-language summary containing the current goal and the verbalization of the latest event. Note that we allow goal nodes to have children of mixed types: either events or other goal nodes. This allows representing subgoals of complex tasks and is used in our real-robot experiments.

L4+ – Higher-Level Summaries: Summaries are generated dynamically by recursively asking an LLM to summarize the previous level's nodes. Specifically, given the set of
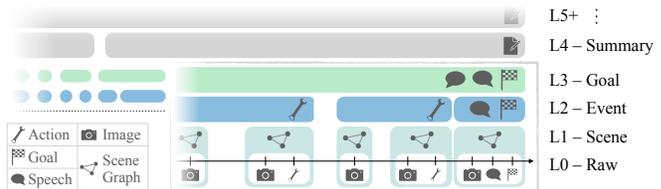

Fig. 2. From the continuous, multimodal stream of robotic experiences, we construct a *history tree*, a hierarchical representation of the EM.

nodes S_ℓ at level $\ell \geq 3$, we list them in order and prompt an LLM to identify consecutive ranges of items that belong together considering their times and content, and provide a summary for each range. The output is parsed to group the child nodes and create the summary nodes $S_{\ell+1}$ for the next level. We apply this strategy recursively, until $|S_\ell| = 1$ or there is no further reduction, i.e., $|S_{\ell+1}| = |S_\ell|$. In the latter, the LLM is explicitly prompted to provide a single concise summary of all items to force obtaining one root node.

B. Episodic Memory Access

Given a user's query and the history tree built from all experiences so far, we use an LLM as an agent [45] to explore the tree, search relevant information, and eventually answer the question. For this, we define an API to interact with the history tree. We initially define each node of the tree to be in a collapsed state, i.e., its textual representation will only contain the node's time range and natural-language summary, but not list the child nodes. The LLM can then interactively expand and collapse nodes, according to what seems relevant given the user's query. Furthermore, we provide different tools to the LLM, e.g., to invoke a Vision-Language-Model (VLM) to perform visual QA on the images associated with leaf nodes. Moreover, there is a function to perform tree search based on semantic similarity, selectively expanding the children of the searched node in the tree that match the search query.

Fig. 3 illustrates typical steps the LLM performs to answer a user's question. Given the initially collapsed tree, the LLM first expands the root node's children based on the requested date. It then selectively explores the respective child nodes that seem relevant to the question using the search function. Note that the LLM is prompted to collapse irrelevant nodes again in order to save token budget and speed up further requests. In the given example, when reaching a leaf node, the answer to the question is not evident from any of the natural-language summaries on each level, so the LLM decides to invoke a VLM to gather more information. Finally, it invokes the `answer` function to answer the user's question.

Our implementation of the LLM agent uses a prompting style inspired by the simulated Python console of [13]. The LLM can issue any command – including compound statements such as loops – using the provided API. After the execution of the respective code, the LLM can "see" the output of its command(s), or any execution error. This process is repeated, and the prompt to the LLM always contains the (growing) execution history. Zero-shot experiments prompt the LLM with only a static prefix to explain the task
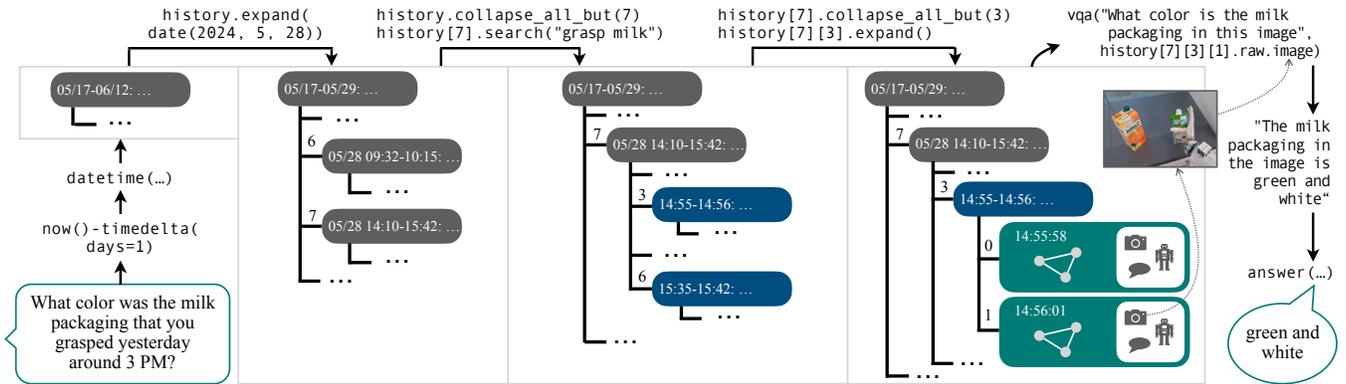
Fig. 3. To answer a user's question, H-EMV prompts an LLM to interactively explore the history tree containing the agent's experiences. The LLM can further invoke tools (index search, VLM) or perform other calculations to gather relevant information, eventually invoking the `answer` function. This figure shows an example from our real-world evaluation on the humanoid robot ARMAR-7, modified for illustrative purposes.

and the available API, while few-shot prompting adds top-k examples selected based on semantic similarity to the current user query. The final part of the prompt is always a string representation of the history tree's current state.

IV. EVALUATION

A. Simulated Household Episodes

Following our previous work [6], [1], we use simulated household episodes and automatically annotate them with QA pairs based on the ground-truth (GT) simulation state. Specifically, we use the TEACh dataset [15], featuring episodes of two real humans, one commander, and one follower, interacting with the AI2THOR environment [46]. We adapt this data by rephrasing the commander to be a human user, and the follower to be a robot interacting with the environment (and the user). Thus, each TEACh episode describes a robot experience, comprising egocentric images, robot states and actions, and dialog with the user.

Data: Episodes in TEACh are on average 6.2 ± 5.3 min long. Since we are interested in very long histories of robot experience, we randomly combine them to form histories of up to 100 episodes. We also randomize dates and times for each episode, ensuring realistic sequences by picking one to five episodes per day, avoiding nighttimes, and occasionally skipping some days; the longest histories thus span nearly two months. Based on these histories, we generate QA pairs by adjusting the generation grammar from [6]. Specifically, we generate ten types of questions. These ask for: a list of high-level summaries of episodes (task descriptions); a detailed description of one particular episode in a history; a summary of an episode that happened either before or after a particular episode; a list of episodes in which a particular object was seen or action performed; a summary of an episode that occurred at a given time or a specified number of days ago; a list of times or number of days ago at which a given task was performed. From the TEACh "valid unseen" set, we generate test sets with 10 histories per sequence length (combining $|h| = 5, 15, 25, 50$, and 100 episodes). Each history is annotated with 10 QA pairs, making up 100 samples per history length.

Evaluation Metrics: Evaluation of free-form EMV answers is hard since there can be many ways to formulate

the correct answer, questions can be underspecified, and verifying abstract statements by grounding them in the history tree is a research question in itself. Following [1], we define a semantic categorization of a model's hypothesis h given the GT g and question q: *correct* when h is semantically equivalent to q; *correctly summarized* if h is a correctly summarized version of g, still containing all relevant facts (in context of q); *correct TMI* (too much information) if h is correct but overly specific; *partially correct TMI* if parts of h are correct, but there are TMI parts and these are wrong; *partially correct missing* if parts of h are correct, but relevant facts from g are missing; *wrong* when h could be an answer to q but is none of the above; and *no answer* if h is empty, completely irrelevant to q, or the model threw an error. Since categorizing each evaluated sample by hand is prohibitively expensive, we prompt GPT-4o [47] to perform this evaluation. For this, we started by annotating 60 samples by hand, and use these as a database to retrieve few-shot samples based on maximal marginal relevance [48]. We further tuned the prompts on a validation set of 100 hand-categorized model outputs. For reporting, we aggregate these semantic categories as the percentage of correct and partially correct samples, S_c and S_p, respectively.

We evaluate the agreement of the LLM's predicted categories with manually annotated ones on 200 model results from our test data, resulting in an aggregated category accuracy of 88%, and per-class f-scores of $F_1(\text{correct}) = 0.89, F_1(\text{partially\_correct}) = 0.84, F_1(\text{wrong}) = 0.91$. The LLM categorizes correct and wrong samples very well and has the most difficulties on the *partially correct* labels. However, these categories are also defined imprecisely, and the inter-annotator agreement [49] between the first two authors has only a value of Cohen's $\kappa = 0.66$ ($n = 110$), vs. $\kappa = 0.91$ ($n = 68$) when only considering correct/wrong. Thus, while not perfect, we use the LLM to automatically obtain reasonable score estimates.

In addition to manual and LLM-predicted categorization, we report standard automated metrics from machine translation that are often applied to free-form QA [6], [18]: BLEU-4 [50] and ROUGE-L (f-score) [51]. However, we emphasize that these surface-level metrics cannot grasp all varieties

→ $|h|$	5					15					25					50					100				
↓ method	B	R	S_c	S_p	T	B	R	S_c	S_p	T	B	R	S_c	S_p	T	B	R	S_c	S_p	T	B	R	S_c	S_p	T
vision-only																									
Gemini 1-pass FS all	26.5	50.3	29	27	325	19.0	34.8	16	27	663	11.6	32.6	17	28	824	5.3	21.4	8	23	1019			OOC		
Gemini 1-pass FS 2nd	30.2	45.2	24	25	164	14.8	36.4	13	30	334	10.6	32.5	17	24	406	3.7	21.8	11	24	512	0.0	20.8	10	20	1256
H-EMV (1-shot)	2.4	18.9	12	24	7.2	2.0	17.9	14	24	13.6	1.9	16.8	16	27	14.2	2.4	17.5	16	24	10.5	0.1	17.6	16	18	16.4
vision + speech																									
Gemini 1-pass FS 2nd	39.5	65.0	57	26	165	37.0	47.5	30	43	336	37.9	48.5	39	30	418	25.2	40.9	25	35	861	16.9	35.2	17	37	1591
H-EMV (1-shot)	4.3	32.9	27	38	11.6	3.9	28.7	36	37	10.5	2.2	30.3	36	35	11.8	5.8	26.3	29	35	10.6	1.0	17.3	24	33	15.1
full multimodal (objects + speech + actions)																									
H-EMV (1-shot)	5.1	34.7	57	18	9.9	4.0	32.0	50	24	10.4	4.7	29.9	46	24	11.4	1.9	27.2	45	28	9.2	1.1	25.2	31	33	10.7
H-EMV (0-shot)	1.2	21.2	48	26	4.8	0.9	18.2	44	25	4.4	0.4	19.4	45	21	4.7	0.1	18.4	43	20	4.8	0.0	15.0	32	28	5.0
H-EMV (0-shot, L3)	0.9	19.9	24	29	25.2	1.0	17.6	18	3	65.6	0.1	16.1	25	24	60.4	0.1	12.1	17	24	82.9	0.0	4.4	4	11	16.6
Gemini 1-pass 0-shot L3	2.4	25.5	32	33	120	1.4	24.7	37	26	372	0.5	27.6	35	34	422	0.2	20.3	24	32	1055	0.0	15.2	3	5	1893

B: BLEU, R: ROUGE, S_c, S_p: semantically categorized (partially) correct in %, T: number of 1K prompt token. Gray token costs exclude out-of-context (OOC) samples.

TABLE I

RESULTS ON SIMULATED HOUSEHOLD EPISODES FROM TEACH [15]

of correct answers in the EMV task, e. g., for a "when"-question, both of the following answers are correct, but have no word overlap at all: "at 4 PM" vs. "in the afternoon".

Settings and Baselines: We evaluate under three settings: First, vision-only can act solely on the visual data stream. As a baseline, we prompt Gemini 1.5 Pro [10] in one pass with the sequence of images along with timestamps and the question. While we use GPT-4o for H-EMV, the 1-pass baseline uses Gemini because it requires extremely long requests. However, despite Gemini's 2M token context length, we need to sample every 2nd frame for longer histories. We few-shot-prompt with one static example history of five episodes including ten QA samples for this history. H-EMV does not take raw images, but constructs history trees by inferring objects using YOLO-World [52] and actions using a LongT5 transformer model [53] fine-tuned on TEACh train. Second, vision + speech enriches the visual information with the dialog data from TEACh episodes, representing natural language commands given to the robot. This is simply added to the prompt for Gemini, and inserted into the history tree for H-EMV. Finally, full multimodal uses the recorded (GT) actions and goals from the TEACh episodes, as this information is typically available when a robot executes some actions. This setting also uses GT object information to compare system performance assuming perfect vision components. We compare H-EMV with one-shot and zero-shot prompting of the LLM agent. For preparing the few-shot samples, we use histories built from episodes in TEACh train, and record traces of manually using the Python console interface and the defined API to interact with the history tree until the given GT answer becomes evident. While we collect two to three samples per question type this way, making up 21 samples in total, we select only the top-1 sample when prompting the LLM, based on semantic similarity of the user's questions. Semantic similarity is determined after asking `gpt-4o-mini` to cross out the task-specific words from the question so that an example from the same question type is retrieved (instead of an irrelevant example just mentioning the same objects or activities). We further ablate the hierarchical summarization: H-EMV *0-shot L3* is our method without LLM-generated summaries (L4+), still using the interactive agent to explore an initially collapsed list of L3 nodes. Last, *Gemini 1-pass 0-shot L3* is a baseline presenting the fully expanded tree (L3 and lower) to the LLM along with the question in a single prompt, thus not using the LLM as an agent.

Results: Results of our TEACh experiments can be found in Table I. First, we can observe that every method's performance decreases with increasing $|h|$. Further, comparing the surface metric scores (B, R) of vision-only 1-pass with full multimodal H-EMV demonstrates that these metrics are not sufficient for evaluating EMV: While vision-only 1-pass has significantly higher B and R, H-EMV actually performs better on the semantic scores. This can be explained by our 1-pass prompting seeing more QA samples and thus better picking up the vocabulary of the generated data, which does not necessarily improve the correctness of the hypotheses but increases n-gram overlap. In contrast, H-EMV using GPT-4o tends to give more free-form answers, especially since it uses hierarchically generated summaries.

Focusing on the vision-only and vision + speech results, the Gemini 1-pass baseline outperforms H-EMV for shorter histories. This is reasonable, as the baseline can directly access the full stream of visual information, whereas our hierarchical system suffers from error propagation and is limited by pretrained vision components. In particular, the history tree could contain incomplete or wrong information or our method could fail by expanding the wrong nodes of the tree, which cannot happen to the 1-pass baseline. However, token costs scale linearly with history length for 1-pass, while it stays approximately constant for H-EMV. The performance also drops faster for 1-pass, with H-EMV reaching comparable or better semantic scores for $|h| \geq 25$.

In contrast, when circumventing the limitations of perception components by using GT object detection and action information (full multimodal setting), H-EMV outperforms the 1-pass system in the semantic metrics, with a token budget two orders of magnitudes smaller. Further, 1-shot prompting significantly helps, but 0-shot also works reasonably well with half the token costs. Ablating the hierarchical higher-level summaries significantly increases token cost and leads to worse performance, also resulting in OOC errors for longer histories (when the LLM expands all nodes).

B. Egocentric Human Videos

Next to verbalizing robot experience, EMV can be applied to human egocentric recordings, e. g., in the context of smart

wearables. Here, the system does not summarize and answer questions about its own actions, but the actions of its user.

Data: To evaluate our system under this setting, we use Ego4D [16]. Randomly concatenating episodes (as done above for TEACh) generates histories that are not cohesive, thus restricting automatic summarization to bare enumeration instead of abstraction of related events. Therefore, we perform a small-scale evaluation on very long recordings from Ego4D. Specifically, we manually select two very long Ego4D videos (6:43h and 4:28h) showing diverse and interesting actions in a tourist scenario. Additionally, we construct one history by concatenating shorter episodes from similar scenarios, selected to ensure some level of cohesiveness and plausibility (in contrast to random sequences). We manually write 40 challenging QA samples.

Method: To construct history trees from Ego4D videos, we apply VideoReCap [9] which produces low-level narrations at 1 fps and mid-level summaries for each minute. We map these to action (L2) and goal (L3) nodes of our hierarchy, respectively, converting texts to first-person perspective using `meta-llama3-8b` [54]. For constructing higher-level (L4+) summaries, we generated few-shot samples for the group-and-summarize LLM (see Sec. III-A) using Ego4D-HCap [9]. To populate the L1 scene graph with objects, we apply YOLO-World [52], an open-vocabulary object detection approach, which we prompt with classes obtained through a Socratic Models [7] approach: First, we select the top-100 classes according to cosine similarity of the mean CLIP [55] image embedding within one L3 node and the CLIP text embeddings of all LVIS [56] labels. Further, we prompt `meta-llama3-8b` to propose ≈ 20 objects that might occur given the current L3 goal annotation produced by VideoReCap. We then apply YOLO-World with the combined set of classes on each image within this L3 node and store the detected objects in the respective L1 nodes. The EMV agent for QA is instantiated zero-shot.

Results: The results of applying our method and manually categorizing the results (following Sec. IV-A) can be seen on the left of Table II. Due to the very challenging nature of our QA samples and the limitations of the used vision components, the overall performance is low. Low performance of the Gemini 1-pass baseline can partially be explained by it seeing only a flat version of the L2 events without access to the images. However, it also fails on most of the samples that could be answered from the text history. This may be explained by the noisy text history inferred from vision, which H-EMV can handle better because of hierarchical summarization and selective expansion of nodes, whereas the 1-pass baseline observes all the noise at once. We did not directly apply the 1-pass baseline on the images (which would be possible only with aggressive sub-sampling) for cost reasons. H-EMV L3 performs similar to H-EMV, with double the costs.

C. Real-World Robot Recordings

Finally, we apply our method on the real-world humanoid robot ARMAR-7. To obtain an EM, we record multiple robot

	Ego4D					ARMAR-7				
	B	R	S_c	S_p	T	B	R	S_c	S_p	T
H-EMV	1.1	11.6	28	25	21.9	7.0	18.6	43	27	12.8
H-EMV (L3)	0.9	13.7	25	18	57.6	2.5	20.4	30	17	68.4
Gemini 1-pass flat	0.8	4.7	5	8	438	12.4	28.8	40	10	227

TABLE II

EXPERIMENTS ON HUMAN & ROBOTIC REAL-WORLD DATA (0-SHOT)

sessions of typical household tasks, spanning a total duration of 3.3 hours of robot actions over the scope of two months. We record all entries made to the memory system introduced in [22], in particular: vision (RGB and depth images), robot state (proprioception), skill events (executed actions and goals), speech (speech-to-text output and text-to-speech input), symbolic scene (objects and their relations). From such recordings, we build up a history tree by populating L0 with images, speech, and proprioception, L1 scene graphs with the symbolic scene information, L2 and L3 with robot action events (where L2 contains low-level actions and L3 contains actions that themselves invoke other actions). Note that L3 nodes can be nested in this case (goals and their subgoals). Higher levels (L4+) are constructed dynamically by an LLM as described in Sec. III-A, with two manually created few-shot samples.

Subsequently, we annotate the recordings with 30 QA-pairs, apply our method, and again manually categorize the results. Results can be seen on the right part of Table II. In general, our task is very challenging, and the 1-pass Gemini baseline which has direct access to the complete stream of episodic data (without images) scores only 40%/10% of correct/partially correct samples. Compared to the Ego4D experiment, the quality of the text history is better, as most content (esp. current action, goal) is not inferred from vision. Our interactive hierarchical system achieves slightly better performance, with 1/17 of the token costs. The numbers also highlight that the hierarchical aspect is crucial, as H-EMV with only L3 has notably lower performance with more than 5 times the token cost. See the supplementary video for a demonstration of our system in action, enabling ARMAR-7 to answer questions about its past interactively.

V. CONCLUSION & DISCUSSION

We present H-EMV, a system for verbalization of life-long robot experience. The multimodal, hierarchical representation of EM is interactively accessed by an LLM to answer user questions, keeping token costs low even for extremely long histories. Despite the promising results and versatility of our system, it has some limitations: First, as a modulated approach, it is limited by the performance of each component and can suffer from error propagation. While the interactive tree search improves interpretability, there are no performance guarantees. Moreover, our system could integrate more modalities and tools. For instance, joint angle proprioception data could be rendered in simulation and then verbalized by a VLM. Adding personalization, both to EM and verbalization, is desirable for improved human-robot interactions. We hope our code and data will foster research on EMV, and will continue addressing these challenges in future work.

REFERENCES

[1] L. Bärmann, F. Peller-Konrad, S. Constantin, T. Asfour, and A. Waibel, "Deep episodic memory for verbalization of robot experience," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5808–5815, 2021.

[2] S. Rosenthal, S. P. Selvaraj, and M. Veloso, "Verbalization: Narration of autonomous robot experience," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 862–868. [Online]. Available: http://dl.acm.org/citation.cfm?id=3060621.3060742

[3] Q. Zhu, V. Perera, M. Wächter, T. Asfour, and M. M. Veloso, "Autonomous narration of humanoid robot kitchen task experience," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 390–397.

[4] C. DeChant and D. Bauer, "Toward robots that learn to summarize their actions in natural language: a set of tasks," in *5th Annual Conference on Robot Learning, Blue Sky Submission Track*, 2021. [Online]. Available: https://openreview.net/forum?id=n3AW_ISWCXf

[5] E. Tulving, "Episodic and semantic memory," *Organization of memory*, vol. 1, pp. 381–403, 1972.

[6] C. DeChant, I. Akinola, and D. Bauer, "Learning to summarize and answer questions about a virtual robot's past actions," *Autonomous Robots*, 2023. [Online]. Available: https://doi.org/10.1007/s10514-023-10134-4

[7] A. Zeng, M. Attarian, B. Ichter, K. M. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. S. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, and P. Florence, "Socratic models: Composing zero-shot multimodal reasoning with language," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=G2Q2Mh3avow

[8] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing robot experiences for failure explanation and correction," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=8yTS_nAILxt

[9] M. M. Islam, N. Ho, X. Yang, T. Nagarajan, L. Torresani, and G. Bertasius, "Video ReCap: Recursive captioning of hour-long videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 198–18 208.

[10] G. Team, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," 2024. [Online]. Available: http://arxiv.org/abs/2403.05530

[11] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg, "RULER: What's the real context size of your long-context language models?" 2024. [Online]. Available: http://arxiv.org/abs/2404.06654

[12] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.9

[13] L. Bärmann, R. Kartmann, F. Peller-Konrad, J. Niehues, A. Waibel, and T. Asfour, "Incremental learning of humanoid robot behavior from natural interaction and large language models," *Frontiers in Robotics and AI*, vol. 11, 2024. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1455375

[14] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "SayPlan: Grounding large language models using 3d scene graphs for scalable robot task planning," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=wMpOMO0Ss7a

[15] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, "TEACh: Task-driven embodied agents that chat," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, pp. 2017–2025, 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20097

[16] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. González, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolář, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. Ruiz, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbeláez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 995–19 012.

[17] T. Asfour, R. Dillmann, N. Vahrenkamp, M. Do, M. Wächter, C. Mandery, P. Kaiser, M. Kröhnert, and M. Grotz, "The karlsruhe armar humanoid robot family," in *Humanoid Robotics: A Reference*. Springer Netherlands, 2017, pp. 1–32.

[18] L. Bärmann and A. Waibel, "Where did i leave my keys? - episodic-memory-based question answering on egocentric videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022, pp. 1560–1568. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022W/Ego4D-EPIC/html/Barmann_Where_Did_I_Leave_My_Keys_-_Episodic-Memory-Based_Question_Answering_CVPRW_2022_paper.html

[19] A. Waibel, H. Steusloff, R. Stiefelhagen, *et al.*, "Chil: Computers in the human interaction loop," 2005.

[20] J. Rothfuss, F. Ferreira, E. E. Aksoy, Y. Zhou, and T. Asfour, "Deep episodic memory: Encoding, recalling, and predicting episodic experiences for robot action execution," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4007–4014, 2018.

[21] C. DeChant, I. Akinola, and D. Bauer, "In search of the embgram: forming episodic representations in a deep learning model," in *Cognitive Computational Neuroscience 2024*, 2024. [Online]. Available: https://2024.ccneuro.org/pdf/141_Paper_authored_CCN_2024_submission_with_names.pdf

[22] F. Peller-Konrad, R. Kartmann, C. R. G. Dreher, A. Meixner, F. Reister, M. Grotz, and T. Asfour, "A memory system of a robot cognitive architecture and its implementation in ArmarX," *Rob. Auton. Sys.*, vol. 164, p. 20, 2023.

[23] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "KnowRob 2.0 — a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. [Online]. Available: https://doi.org/10.1109%2Ficra.2018.8460964

[24] J. Plewnia, F. Peller-Konrad, and T. Asfour, "Forgetting in robotic episodic long-term memory," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6711–6717.

[25] A. Yuguchi, S. Kawano, K. Yoshino, C. T. Ishi, Y. Kawanishi, Y. Nakamura, T. Minato, Y. Saito, and M. Minoh, "Butsukusa: A conversational mobile robot describing its own observations and internal states," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022, pp. 1114–1118.

[26] M. Han, M. Kang, H. Jung, and S. J. Hwang, "Episodic memory reader: Learning what to remember for question answering from streaming data," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4407–4417. [Online]. Available: https://aclanthology.org/P19-1434

[27] V. Araujo, A. Soto, and M.-F. Moens, "A memory model for question answering from streaming data supported by rehearsal and anticipation of coreference information," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 13 124–13 138. [Online]. Available: https://aclanthology.org/2023.findings-acl.830

[28] Y. Zhong, W. Ji, J. Xiao, Y. Li, W. Deng, and T.-S. Chua, "Video question answering: Datasets, algorithms and challenges," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 6439–6455. [Online]. Available: https://aclanthology.org/2022.emnlp-main.432

[29] Y. Song, G. Byrne, T. Nagarajan, H. Wang, M. Martin, and L. Torresani, "Ego4d goal-step: Toward hierarchical understanding of procedural activities," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. [Online]. Available: https://openreview.net/forum?id=3BxYAaovKr

[30] K. Mangalam, R. Akshulakov, and J. Malik, "EgoSchema: A diagnostic benchmark for very long-form video language understanding," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. [Online]. Available: https://openreview.net/forum?id=JVlWseddak

[31] I. Balažević, Y. Shi, P. Papalampidi, R. Chaabouni, S. Koppula, and O. J. Hénaff, "Memory consolidation enables long-context video

understanding," 2024. [Online]. Available: http://arxiv.org/abs/2402.05861

[32] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang, Y. Lu, J.-N. Hwang, and G. Wang, "MovieChat: From dense token to sparse memory for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 221–18 232.

[33] R. Tan, X. Sun, P. Hu, J.-h. Wang, H. Deilamsalehy, B. A. Plummer, B. Russell, and K. Saenko, "Koala: Key frame-conditioned long video-LLM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 13 581–13 591.

[34] S. Di and W. Xie, "Grounded question-answering in long egocentric videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 12 934–12 943.

[35] Y. Fan, X. Ma, R. Wu, Y. Du, J. Li, Z. Gao, and Q. Li, "VideoAgent: A memory-augmented multimodal agent for video understanding," 2024. [Online]. Available: http://arxiv.org/abs/2403.11481

[36] Z. Wang, S. Yu, E. Stengel-Eskin, J. Yoon, F. Cheng, G. Bertasius, and M. Bansal, "VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos," 2024. [Online]. Available: http://arxiv.org/abs/2405.19209

[37] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy, "VideoAgent: Long-form video understanding with large language model as agent," 2024. [Online]. Available: http://arxiv.org/abs/2403.10517

[38] J. Min, S. Buch, A. Nagrani, M. Cho, and C. Schmid, "MoReVQA: Exploring modular reasoning models for video question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 13 235–13 245.

[39] Y. Wang, Y. Yang, and M. Ren, "LifelongMemory: Leveraging LLMs for answering queries in egocentric videos," 2023. [Online]. Available: http://arxiv.org/abs/2312.05269

[40] R. Choudhury, K. Niinuma, K. M. Kitani, and L. A. Jeni, "Zero-shot video question answering with procedural programs," 2023. [Online]. Available: http://arxiv.org/abs/2312.00937

[41] C. Zhang, T. Lu, M. M. Islam, Z. Wang, S. Yu, M. Bansal, and G. Bertasius, "A simple LLM framework for long-range video question-answering," 2024. [Online]. Available: http://arxiv.org/abs/2312.17235

[42] X. Wang, Y. Chen, L. Yuan, Y. Zhang, Y. Li, H. Peng, and H. Ji, "Executable code actions elicit better LLM agents," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=jJ9BoXAfFa

[43] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou, "TimeChat: A time-sensitive multimodal large language model for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 313–14 323.

[44] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *it - Information Technology*, vol. 57, 2015.

[45] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024. [Online]. Available: https://link.springer.com/10.1007/s11704-024-40231-1

[46] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[47] OpenAI, "GPT-4 Technical Report," *arXiv:2303.08774*, 2023.

[48] X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett, and R. Pasunuru, "Complementary explanations for effective in-context learning," in *ACL*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 4469–4484.

[49] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[50] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002, pp. 311–318.

[51] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. ACL, 2004, pp. 74–81.

[52] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 16 901–16 911.

[53] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang, "LongT5: Efficient text-to-text transformer for long sequences," in *Findings of the Association for Computational Linguistics: NAACL 2022*, July 2022, pp. 724–736. [Online]. Available: https://aclanthology.org/2022.findings-naacl.55

[54] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: http://arxiv.org/abs/2407.21783

[55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[56] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.