

Look, Remember and Reason: Grounded Reasoning in Videos with Language Models

Apratim Bhattacharyya, Sunny Panchal, Mingy Lee, Reza Pourreza, Pulkit Madan, Roland Memisevic
Qualcomm AI Research*

Abstract

Multi-modal language models (LM) have recently shown promising performance in high-level reasoning tasks on videos, but struggle with causal or compositional reasoning that requires fine-grained, low-level visual understanding. To address this, we propose training an LM on low-level surrogate tasks such as object detection, re-identification, and tracking. This endows the model with essential visual capabilities. Further, we use a two-stream video encoder with spatiotemporal attention to effectively capture the required static and motion-based cues in videos. Our framework reframes video reasoning as Look, Remember, Reason, where our model extracts low-level visual information step-by-step for final reasoning. Evaluation on ACRE, CATER, Something-Else, and STAR datasets demonstrates that our approach outperforms task-specific methods across visual reasoning tasks.

1. Introduction

Drawing from the success of autoregressive language models (LMs) on text based reasoning tasks, multi-modal LMs for images or videos have recently gained traction. The focus of multi-modal LMs for videos have primarily been on high-level question answering and instruction following. However, many visual reasoning problems in videos require grounding in fine-grained low-level information, *i.e.*, recognizing objects, and understanding their spatiotemporal interactions. The ability of multi-modal LMs to perform visual reasoning tasks such as compositional action recognition in videos [12] that require a combination of low-level skills with high-level reasoning has not yet been explored.

To enable multi-modal LMs to solve such reasoning problems we propose our *Look, Remember and Reason (LRR)* multi-modal LM. Our LRR model architecture extracts dense spatiotemporal features from each input frame. This is accomplished using a two-stream attention-based architec-

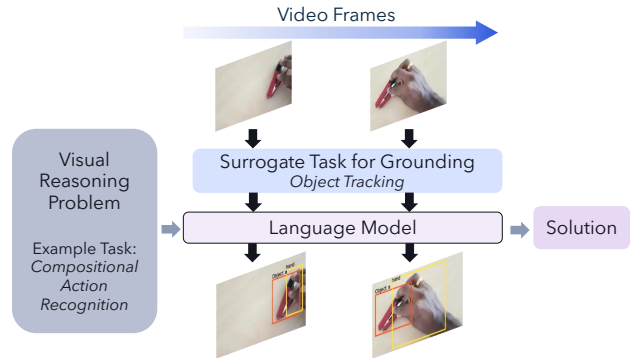


Figure 1. Our Look, Remember, Reason (LRR) model ‘looks’ at the video frames to extract relevant low-level information, *e.g.*, object motion and interactions, supervised with surrogate tasks. It ‘remembers’ the information from intermediate steps and ‘reasons’ using the aggregated information.

ture that captures low-level spatial and temporal details [14] at each input video frame using top-down cross-attention. Our multi-modal LRR model is grounded to relevant low-level visual information in the scene by stochastically introducing low-level surrogate tasks during training, including object recognition, re-identification, and tracking, at randomly selected time-steps (*c.f.* Fig. 1). We keep these grounded spatiotemporal features, which include low-level visual details, in the working memory, *i.e.*, “remembered” within the context window of the LM. This allows the model to combine the low-level visual features with high-level inferences to “reason” and generate the final responses using our LRR framework as shown in Fig. 1.

Our main contributions are: 1) We highlight the importance of grounding for visual reasoning in multi-modal LMs and propose a novel Look, Remember, and Reason framework to this end to instill the required low-level visual skills in the model using surrogate tasks during training; 2) We introduce a two-stream video encoder that captures both the scene structure and object motions, crucial for learning the low-level skills; 3) We demonstrate the effectiveness of our approach on ACRE [19], CATER [7], and the real-world Something-Else [12] and STAR [17] datasets. Our approach

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

outperforms the prior state-of-the-art, that is based on highly task-specific architectures, by a large margin—highlighting how our general-purpose LRR model can perform varied and complex spatiotemporal reasoning tasks in videos including causal, compositional and situated reasoning.

2. Look, Remember, Reason

In the following, we first describe our auto-regressive LRR architecture including details of our two-stream video encoder, followed by our training pipeline with details of our surrogate tasks.

2.1. Auto-regressive Pipeline

Our LRR model parameterized by θ , as shown in Fig. 2, is based on a pre-trained LM backbone with a two-stream auto-regressive video encoder. Our LRR model receives the visual reasoning problem \mathbf{Q} and, an interleaved sequence \mathcal{I} of video frames $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{T_v})$ and tokenized text $\mathbf{S} = (\mathbf{s}_1, \dots, \mathbf{s}_{T_s})$ as input. \mathbf{S} consists of low-level visual surrogate tasks and the answer indicated by $\langle \text{taskname} \rangle$ and $\langle \text{answer} \rangle$ respectively. The input video frame sequence \mathbf{V} is encoded by our two-stream video encoder. The LM backbone receives as input \mathbf{Q} , and the interleaved sequence of encoded video frames \mathbf{V} whose positions are indicated with $\langle \text{frame} \rangle$ special tokens and \mathbf{S} . Conditioned on \mathbf{Q} and \mathbf{V} , we train our LRR model by maximizing log-likelihood of the text sequence \mathbf{S} as,

$$\log(p_\theta(\mathbf{S}|\mathbf{Q}, \mathbf{V})) = \sum_{t_s} \log(p_\theta(\mathbf{s}_{t_s}|\mathbf{s}_1, \dots, \mathbf{s}_{t_s-1}, \mathbf{v}_1, \dots, \mathbf{v}_{t_v}, \mathbf{Q})) \quad (1)$$

where, $(\mathbf{v}_1, \dots, \mathbf{v}_{t_v})$ is the interleaved video frame subsequence up to the text token \mathbf{s}_{t_s} . In Sec. 2.3, we describe the surrogate tasks included in \mathbf{S} . The parameters of the LM backbone are initialized from pre-trained LMs, allowing us to exploit their existing high-level reasoning capabilities.

2.2. Two-stream Video Encoder

Our autoregressive two-stream video encoder exploits divided space-time attention [3] and generates for each input video frame patch based embeddings that capture relevant low-level spatio-temporal information. Spatial attention captures structural information, *e.g.*, object identities. Temporal attention autoregressively captures object motion and interaction information using the previous τ frames as a buffer. As a first step, it converts each input video frame \mathbf{v}_{t_v} into P flattened patches $\mathbf{v}_{t_v} = \{\mathbf{v}_{(1,t_v)}, \dots, \mathbf{v}_{(P,t_v)}\}$ of size 16×16 [5] as shown in Fig. 2. Our LRR model applies a linear transformation (LN) on the input patches to generate the initial patch embeddings $\mathbf{z}_{t_v}^0 = \{\mathbf{z}_{(1,t_v)}^0, \dots, \mathbf{z}_{(P,t_v)}^0\}$ as,

$$\mathbf{z}_{(p,t_v)}^0 = \text{LN}(\mathbf{v}_{(p,t_v)}) + \mathbf{em}_{(p,t_v)} \quad (2)$$

The spatiotemporal positional embedding $\mathbf{em}_{(p,t_v)}$ is added to each patch to aid our video encoder extract spatiotemporal features from each patch in the input video frame. Next, these patch embeddings are processed by the L blocks of our video encoder where at each block spatial or temporal attention is applied. At every block $\ell \in \{1, \dots, L\}$ of our model, the embeddings are linearly mapped to produce the query, keys, and values required for the space- and time-based attention operations,

$$\begin{aligned} \mathbf{q}_{(p,t_v)}^\ell &= \text{LN}(\mathbf{z}_{(p,t_v)}^\ell), \quad \mathbf{k}_{(p,t_v)}^\ell = \text{LN}(\mathbf{z}_{(p,t_v)}^\ell), \\ \mathbf{v}_{(p,t_v)}^\ell &= \text{LN}(\mathbf{z}_{(p,t_v)}^\ell). \end{aligned} \quad (3)$$

In case of temporal attention, for the patch embedding $\mathbf{z}_{(p,t_v)}^\ell$ the attention is computed over patch embeddings at the same spatial position p in the τ previous input video frames, $\mathbf{z}_{(p,(t_v-\tau))}^\ell$ to $\mathbf{z}_{(p,t_v)}^\ell$. For the patch highlighted in red in Fig. 2 the patches where temporal attention is applied in the previous τ input frames is highlighted in cyan. The temporal attention vector $(\alpha_T)_{(p,t_v)}^\ell$ is given by,

$$\begin{aligned} (\alpha_T)_{(p,t_v)}^\ell &= \\ \text{SOFTMAX}\left(\frac{\mathbf{q}_{(p,t_v)}^{\ell\top}}{\sqrt{d_m}}[\mathbf{k}_{(p,(t_v-\tau))}^\ell, \dots, \mathbf{k}_{(p,t_v)}^\ell]\right) \end{aligned} \quad (4)$$

where d_m is the dimensionality of the key, queries and values. In contrast, the spatial attention vector, $(\alpha_S)_{(p,t_v)}^\ell$, is calculated over all patch embeddings of the current input video frame, highlighted in green in Fig. 2,

$$\begin{aligned} (\alpha_S)_{(p,t_v)}^\ell &= \\ \text{SOFTMAX}\left(\frac{\mathbf{q}_{(p,t_v)}^{\ell\top}}{\sqrt{d_m}}[\mathbf{k}_{(1,t_v)}^\ell, \dots, \mathbf{k}_{(P,t_v)}^\ell]\right) \end{aligned} \quad (5)$$

The patch embeddings $\mathbf{z}_{(p,t_v)}^{\ell+1}$ are the weighted sums of the values $\mathbf{v}_{(p,t_v)}^\ell$ computed using the attention weights $(\alpha_T)_{(p,t_v)}^\ell$ or $(\alpha_S)_{(p,t_v)}^\ell$ in case of temporal or spatial attention, respectively. Finally, the patch embeddings $\mathbf{z}_{t_v}^L$, containing localized information of the scene structure and motion, are obtained as output from the last block L at time-step t_v of the video encoder.

2.3. Grounding Through Surrogate Tasks

Visual information $(\mathbf{z}_{(p,t_v)}^{\ell+1})$ encoded by the two-stream video encoder is mapped top-down to the LM backbone at positions indicated by the $\langle \text{frame} \rangle$ token at time-steps t_v (Fig. 2) using cross attention layers. To this end, we modify the backbone LM architecture by inserting cross attention (CROSS-ATTN) layers [1] between self-attention (SELF-ATTN) layers (Fig. 2).

To ground our LRR model to the relevant low-level information in the visual input, we utilize surrogate tasks during training. This is illustrated in Fig. 2 where the model

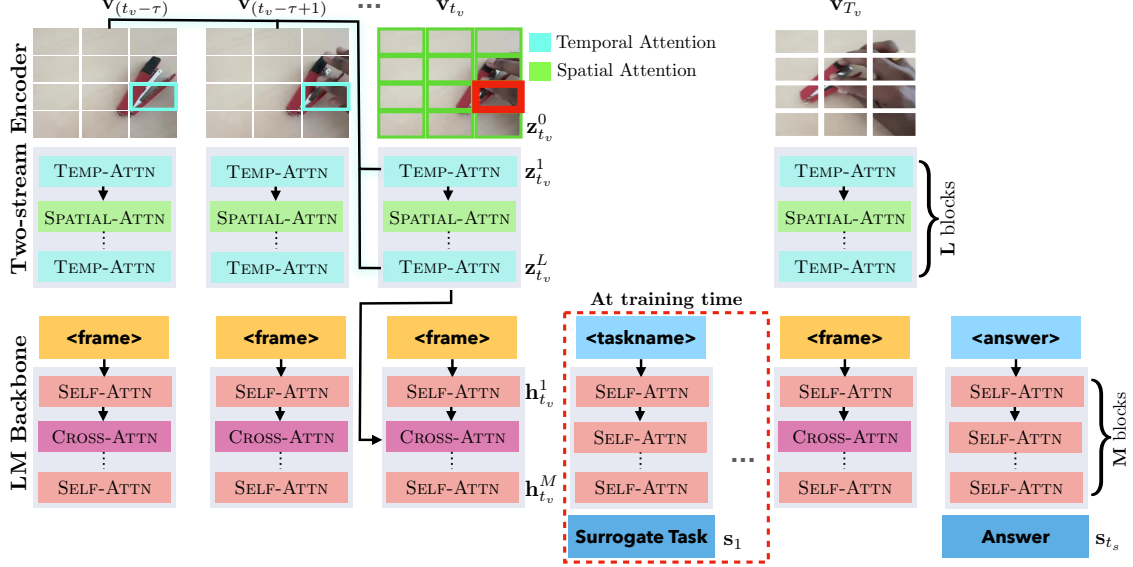


Figure 2. The architecture of our LRR model, highlighting the use of interleaved top-down cross-attention layers in between self-attention layers higher up in the hierarchy.

is prompted using special tokens ($\langle \text{taskname} \rangle$) to solve a surrogate task. We consider tasks like object recognition, tracking and re-identification as low-level. These tasks are fundamental to solving a range of visual reasoning problems, and the requisite ground truth can be readily obtained using off-the-shelf vision models.

Our LRR model is highly flexible and can be prompted to solve a wide range of low-level surrogate tasks by encoding these tasks as text to be processed by the LM backbone in the following general format: “ $\langle \text{taskname} \rangle$ *object id*₁, *object class*₁, *object bounding box*₁; ... ; *object id*_n, *object class*_n, *object bounding box*_n”, where there are n objects in the scene. The $\langle \text{taskname} \rangle$ special token prompts the model to solve the surrogate task. We use $\langle \text{detect} \rangle$, $\langle \text{re-identify} \rangle$ and $\langle \text{track} \rangle$ for detection, re-identification, and tracking surrogate tasks respectively. The *object id* is an integer that is assigned by the model based on the spatiotemporal order of appearance and is crucial for re-identification and tracking as same object instance should be assigned the same id across video frames. The bounding box is described as a 4-tuple of the x and y coordinates of the upper left and lower right corners. We provide additional details in the experimental section (Sec. 3).

Our LRR model is prompted to solve these surrogate tasks at randomly selected time steps during training. Random prompting forgoes the need to include surrogate tasks during inference time, leading to faster inference. Random prompting also benefits training efficiency in the case of long video sequences. To train our LRR model using maximum likelihood as described in Eq. (1). We construct the token sequence \mathbf{S} in \mathcal{I} that includes surrogate tasks for each train-

Method	Int.	Seq.	Pre.	Fea.	Overall \uparrow
Internvideo [16]	62.7	65.6	54.9	51.9	58.7
BLIP-2 [10]	65.4	69.0	59.7	54.2	62.0
SeViLA [18]	63.7	70.4	63.1	62.4	64.9
LRR (Ours)	73.7	71.0	71.3	65.1	70.5
LRR (w/o Surr-tasks)	54.5	48.7	44.3	45.5	48.2

Table 1. Evaluation of our LRR model on STAR (validation set).

ing example as follows: We first randomly select a subset of video frames $(\mathbf{v}_{t_1}, \dots, \mathbf{v}_{t_k}) \in \mathbf{V}$. We then update the token sequence \mathbf{S} to include the surrogate tasks at these randomly selected time-steps, the beginning of which is marked by a task specific $\langle \text{taskname} \rangle$ special token, e.g., $\langle \text{detect} \rangle$, $\langle \text{re-identify} \rangle$ and $\langle \text{track} \rangle$. The surrogate task itself is added in \mathbf{S} as described above.

3. Experiments

Here, we evaluate on STAR [17] and on ACRE [19], Something-Else [12], CATER [7] in the supplemental.

Models and Training Details. We fine-tune the pre-trained LM backbone along with the video encoder and cross-attention layers in our LRR model as the visual reasoning problems considered here are challenging and cannot be accurately solved by prompting state of the art LMs such as GPT-4 [6]. We focus on the OPT family of LMs [21], particularly OPT-125M/350M/1.3B. We use 4 Nvidia A100 GPUs.

3.1. STAR

STAR [17] is a situated spatio-temporal reasoning benchmark, consisting of questions built upon real-world videos associated with human actions and interactions. The STAR benchmark thus requires an understanding of low-level human motion, actions and object interactions. However, the STAR dataset does not contain dense object annotations, *e.g.*, object tracks, unlike the CATER and Something-Else datasets. It only contains sparse spatio-temporal scene graphs which cover selected keyframes. We introduce object recognition surrogate tasks to localize objects on these keyframes based on the scene graphs. Furthermore, we jointly train our LRR model to recognize actions from the Kinetics [9] and Moments in Time [13] dataset, as well as on surrogate tracking tasks from the Something-Else dataset to improve the understanding of object interactions. We also regularize on textual data.

Our LRR model with an OPT-350M LM backbone achieves 70.5% overall accuracy and significantly outperforms powerful transformer-based state of the art models, such as SeViLA [18], Internvideo [16] and BLIP-2 [10] as shown in Tab. 1. Following SeViLA [18], we report results on the validation set. Note that methods such as SeViLA are trained on a much larger set of image/video - text pairs and, use the 4.1B parameter BLIP as a video encoder and the 3B parameter Flan-T5 XL as a language model (*c.f.* Section 4.4 in [18]). In contrast, our LRR model contains fewer parameters and is trained on a much smaller training set. Crucially, our training set includes carefully selected surrogate tasks to endow the model with the requisite low-level visual capabilities. To illustrate this, we include an ablation without any surrogate tasks (w/o Surrogate tasks).

In addition to the results on the STAR validation set presented above we also evaluate our LRR model on the STAR challenge leaderboard. The results can be found: [here](#) (ranked 1st as of February 2024.)

4. Conclusion

We show that off-the-shelf LMs can solve complex visual reasoning tasks on videos using our LRR framework. We equip LMs with a two-stream video encoder and are grounded using surrogate tasks. Grounding ensures that the LM can utilize relevant low-level visual cues from in the input video. Grounding predictions to low-level visual cues combined with the high-level reasoning ability of the LM is the key to the success of the model. Our LRR model outperforms the state of the art by 6.5% and 5.3% on the compositional and systematic splits of the ACRE dataset; by 5.8% Top-1 accuracy on the compositional split of the Something-Else dataset; by 4.4% Top-1 accuracy on static camera and 20.7% Top-1 accuracy on moving camera splits of the CATER dataset; and by 5.6% overall accuracy on STAR.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 2
- [2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *CoRR*, abs/2308.01390, 2023. 5, 6
- [3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2
- [4] David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt M. Botvinick. Attention over learned object embeddings enables complex visual reasoning. In *NeurIPS*, 2021. 5, 6, 7
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2
- [6] Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not abstract reasoners. *CoRR*, abs/2305.19555, 2023. 3
- [7] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for compositional actions & temporal reasoning. In *ICLR*, 2020. 1, 3, 5, 7
- [8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yanilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thureau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 5
- [9] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 4
- [10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 3, 4
- [11] Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *CoRR*, abs/2306.05424, 2023. 5, 7
- [12] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *CVPR*, 2020. 1, 3, 5, 7
- [13] Mathew Monfort, Carl Vondrick, Aude Oliva, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa M. Brown, Quanfu Fan, and Dan Gutfreund. Moments in time dataset: One million videos for event understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):502–508, 2020. 4
- [14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1
- [15] Manuel Traub, Sebastian Otte, Tobias Menge, Matthias Karlbauer, Jannik Thümmel, and Martin V. Butz. Learning what and where - unsupervised disentangling location and identity tracking. In *ICLR*, 2023. 6
- [16] Yi Wang, Kunchang Li, Yizhuo Li, Yanan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *CoRR*, abs/2212.03191, 2022. 3, 4
- [17] Bo Wu, Shoubin Yu, Zhenfang Chen, Josh Tenenbaum, and Chuhan Gan. STAR: A benchmark for situated reasoning in real-world videos. In *NeurIPS*, 2021. 1, 3, 4
- [18] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *CoRR*, abs/2305.06988, 2023. 3, 4
- [19] Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. ACRE: abstract causal reasoning beyond covariation. In *CVPR*, 2021. 1, 3, 5, 6
- [20] Shiwen Zhang. Tfnet: Temporal fully connected networks for static unbiased temporal reasoning. *CoRR*, 2022. 6, 7
- [21] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuo-hui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. 3
- [22] Honglu Zhou, Asim Kadav, Farley Lai, Alexandru Niculescu-Mizil, Martin Renqiang Min, Mubbasir Kapadia, and Hans Peter Graf. Hopper: Multi-hop transformer for spatiotemporal reasoning. In *ICLR*, 2021. 6, 7

Appendix

A. Overview

Here we provide additional evaluation on the ACRE [19], Something-Else [12], CATER [7] datasets.

B. Experiments

B.1. ACRE

The ACRE dataset [19] evaluates how well vision systems perform causal induction. It focuses on causal discovery through "blicket" detection tests (originally designed for children). A blicket detector activates in the presence of a blicket object. The experiment uses context trials where different objects are placed on the detector, revealing its activation status. Subjects must then determine which objects or combinations would trigger the detector. The ACRE dataset includes 6 context trials and 4 blicket detection tests per sample.

Surrogate tasks. The key low-level visual challenge in the ACRE dataset is to associate query objects to the context trials to detect whether the blicket machine is activated. Therefore, we consider the surrogate tasks of object recognition and re-identification. The solution to the surrogate task can be generated by the backbone LM in the following format: "`<re-identify> object id1, object class1; ... ; object idn, object classn`" as shown in Fig. 3. We also found it helpful to introduce an additional surrogate task to identify the state of the blicket machine: "`<blicket> on/off`". These surrogate tasks are introduced randomly during training with a probability of 30% after each context trial or query.

Baselines and evaluation. We base our LRR models on the OPT-125M backbone. For comparison (see Tab. 2), we include the state-of-the-art multi-modal LLM OpenFlamingo (3B-mosaicml/mpt-1b-redpajama-200b-dolly; [2]), as it has shown success in reasoning problems involving multiple images and videos. We also test the following ablations to highlight key components: 1) LRR (w/o Surrogate tasks): Tests the importance of the surrogate re-identification task; 2) LRR (w/o Two-stream encoder): Tests the importance of our two-stream encoder’s temporal attention mechanism for object re-identification; 3) LRR (trained from scratch): Demonstrates the value of pre-trained LM backbones by using the same OPT-125M architecture without pre-training.

Our LRR model outperforms the state of the art powerful transformer based ALOE [4], by a large margin of 6.5% and 5.3% on the compositional and systematic splits respectively. This shows the advantage of our end-to-end LRR model with surrogate tasks over explicit object centric input representations used by ALOE. Similarly, we observe a significant performance gain of 60% over the powerful transformer based multi-modal LLM OpenFlamingo [2] which highlights the importance of our LRR framework to ground

our model to the relevant low-level details – highlighted by the weak performance of the LRR (w/o Surrogate tasks) ablation. We also see that our two-stream encoder improves performance over the single-stream model. The weak performance of the LRR (from scratch) ablation shows that it is crucial to start from a pre-trained LM backbone to exploit its high level reasoning abilities. Finally, although the results in Tab. 2 use the OPT-125M backbone in our LRR model, we obtain an identical 98.2% and 99.0% accuracy on the compositional and systematic splits respectively with an OPT-1.3B backbone, indicating that our *look, remember, reason* framework is applicable across LM backbone sizes.

B.2. Something-Else

This complex, real-world dataset [12] focuses on compositional action recognition. Building upon the Something-Something dataset [8], it measures compositional generalization by splitting actions into verb, subject, and object combinations. This split allows for benchmarking performance on novel combinations unseen during training, forcing models to develop a nuanced understanding of motion rather than simply associating actions with object types.

Surrogate tasks. In this task, we employ tracking as a surrogate task to support the model’s ability to capture motion and object interactions. Note that, as object classes are not important for compositional action recognition, the solution to the surrogate task can be generated in the following format: "`<track> object id1, object bounding box1; ... ; object idn, object bounding boxn`" as shown in Fig. 4. Surrogate tasks are introduced randomly during training with a probability of 30% after each input video frame.

Baselines and evaluation. We base our LRR model on the OPT-125M LM backbone and compare to several baselines and ablations in Tab. 3. We report results on both the base split and the compositional split with novel action-object combinations. We consider the state-of-the-art STIN + OIE + NL [12] and Video-ChatGPT [11] as baselines. We demonstrate the importance of our surrogate tasks and two-stream video encoder through ablations: LRR (w/o Surrogate Tasks) and (w/o Two-stream encoder).

Our LRR model also outperforms the STIN + OIE + NL baseline by 2.1% and 5.8% Top-1 accuracy on the base and compositional split respectively, which highlights the reasoning ability of grounded LM based architectures. The results also show that the performance of the state of the art multi-modal LM: Video-ChatGPT (finetuned), lags very significantly by 23.4% behind our LRR model. This is because of the CLIP based video encoder in Video-ChatGPT is not well suited for capturing motion features and due to a lack of grounding. The importance of motion features is underscored by the lacking performance of the plain ViT based (w/o Two-stream encoder) ablation: a Top-1 accuracy drop of 8.4% on the compositional split. The importance

Model	Compositional					Systematic				
	All	D.R.	I.D.	S.O.	B.B.	All	D.R.	I.D.	S.O.	B.B.
NS-OPT [19]	69.0	92.5	76.0	88.3	13.4	67.4	94.7	88.3	82.7	16.0
ALOE [4]	91.7	97.1	90.8	96.8	78.8	93.9	97.1	71.2	98.9	94.4
OpenFlamingo* [2]	38.2	42.6	49.5	9.9	47.6	38.6	36.5	25.8	13.7	67.6
LRR (Ours)	98.2	99.9	92.8	99.2	97.4	99.2	99.9	97.0	99.9	98.8
LRR (w/o Surrogate tasks)	38.1	38.4	30.2	26.2	50.0	36.5	35.2	28.1	20.0	55.3
LRR (w/o “Two-stream” encoder)	92.2	98.6	77.7	96.2	85.8	92.8	98.5	76.2	96.8	89.4
LRR (from scratch)	87.7	96.9	57.7	91.1	84.8	88.0	96.3	58.5	91.5	86.9

Table 2. Evaluation results on the ACRE dataset, where, D.R. – Direct evidence, I.D. – Indirect evidence, S.O. – Screened-off, and B.B. – Backward Blocked subsets (*represents results tested by ourselves.).

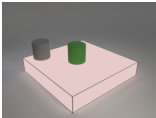
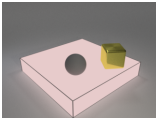
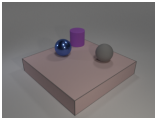
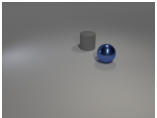
Context Trials			Query
			
<re-identify> <u>1, medium gray rubber cylinder</u> ; 2, medium green rubber cylinder. <blicket> on.	<re-identify> <u>3, medium gray rubber sphere</u> ; 4, medium yellow metal cube. <blicket> on.	<re-identify> <u>5, medium blue metal sphere</u> ; 6, medium purple rubber cylinder; 3, medium gray rubber sphere. <blicket> off.	<re-identify> <u>1, medium gray rubber cylinder</u> ; <u>5, medium blue metal sphere</u> . <answer> no.

Figure 3. Example solutions to surrogate tasks generated by our LRR model on ACRE. Re-identified objects across context trials are underlined in the same color.

of grounding is highlighted through our ablation without surrogate tasks: a Top-1 accuracy drop of 11.9% on the compositional split. We illustrate successful tracking of objects in complex real-world scenarios by our LRR model in Fig. 4. Note that, although the results in Tab. 3 uses the OPT-125M backbone, we obtain an identical 61.3% Top-1 and 85.9% Top-5 accuracy on the compositional split with an OPT-1.3B backbone, indicating that our *look, remember, reason* framework is applicable across LM sizes.

B.3. CATER

The CATER (Compositional Actions and TEmporal Reasoning) dataset challenges models to recognize complex object movement patterns requiring long-term reasoning. The most difficult task is adversarial target tracking, where a “snitch” object must be located at the end of a video sequence despite potential occlusion or containment within other objects. This is formulated as a classification problem on a 6×6 grid. The dataset includes static and moving camera splits, with the latter posing additional challenges due to the need for long-term spatiotemporal analysis.

Surrogate tasks. We employ (multi-target) tracking as a surrogate task for which the solution can be expressed in the following format: “<track> *object id*₁, *object grid position*₁; ... ; *object id*_n, *object grid position*_n” as shown in Fig. 5. The tracking task includes the medium and large cones in the scene, as these objects can occlude the snitch. We use grid positions instead of bounding boxes (unlike Something-Else) because the final goal is to predict the grid position of

the snitch. Surrogate tasks are introduced randomly during training with a probability of 30% after each input frame.

Baselines and evaluation. Our model uses the OPT-125M backbone (results in Tab. 4). We ablate both the surrogate (multi-target) tracking task (w/o Surrogate tasks) and our Two-stream video encoder (w/o Two-stream Encoder) to demonstrate their importance. Following ALOE [4], we jointly train on static and moving camera splits.

The state-of-the-art powerful transformer-based Hopper [22], TFC V3D Depthwise [20] and Loci [15] report results only on the static camera split. Loci [15] reports an impressive 90.7% accuracy on the static camera split, but it is not applicable to the moving camera split due to its static background and camera model. Our LRR model outperforms TFC V3D Depthwise [20] model on the static camera by 4.4% and ALOE [4] on the challenging moving camera split by 20.7%. The large performance gain over the LRR (w/o Surrogate tasks) baseline shows the advantage of surrogate tracking tasks, without which the model is not grounded to the motion of the cones and hence fails in cases where the snitch is contained by the cones. Qualitative examples in Fig. 5 illustrates that our model is able to successfully track objects in cases of recursive containment and is robust to moving cameras. Finally, the performance advantage over the LRR (w/o Two-stream encoder) confirms that our LRR model is able to better capture the motion of the objects.

Method	Base		Compositional	
	Top-1	Top-5	Top-1	Top-5
STIN + OIE + NL [12]	78.1	94.5	56.2	81.3
Video-ChatGPT* [11]	52.6	75.8	38.6	67.8
LRR (Ours)	80.2	96.1	62.0	86.3
LRR (w/o Surrogate tasks)	71.3	89.6	50.1	70.8
LRR (w/o Two-stream encoder)	73.2	90.4	53.6	76.1

Table 3. Evaluation on the Something-Else dataset (* represents results tested by ourselves.).

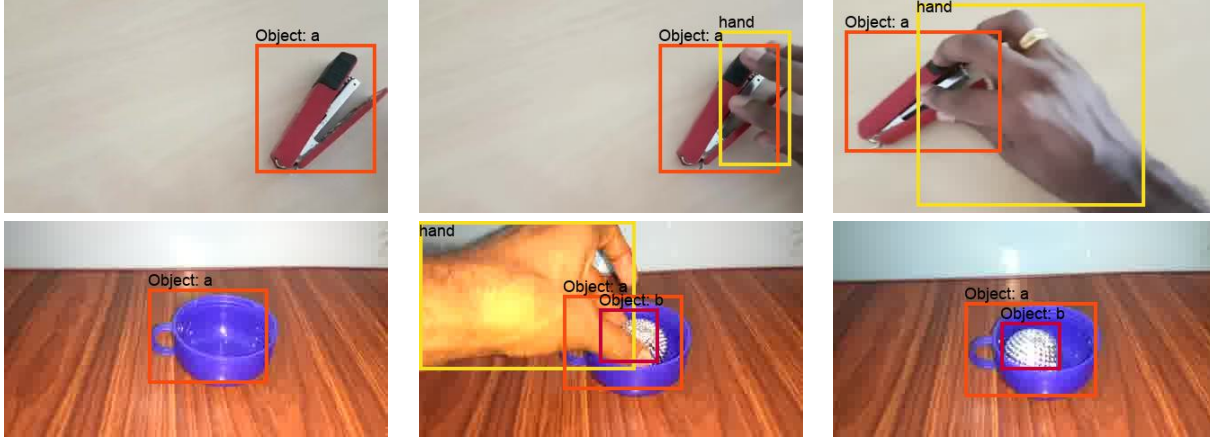


Figure 4. Example solutions to surrogate task tracking generated by our LRR model on Something-Else. Bounding boxes belonging to the same track are highlighted using the same color.

Method	Static Camera			Moving Camera		
	Top-1(↑)	Top-5(↑)	L1(grid;↓)	Top-1(↑)	Top-5(↑)	L1(grid;↓)
R3D + NL LSTM [7]	46.2	69.9	1.5	38.6	70.2	1.5
ALOE [4]	74.0	94.0	0.44	59.7	90.1	0.69
Hopper [†] [22]	73.2	93.8	0.85	-	-	-
TFC V3D [†] [20]	79.7	95.5	0.47	-	-	-
LRR (Ours)	84.1	97.2	0.34	80.4	96.7	0.42
LRR (w/o Surrogate tasks)	68.5	88.7	0.65	62.7	86.7	0.77
LRR (w/o Two-stream encoder)	81.4	97.2	0.44	75.6	96.6	0.53

Table 4. Evaluation on the CATER dataset ([†] results reported only for static camera).

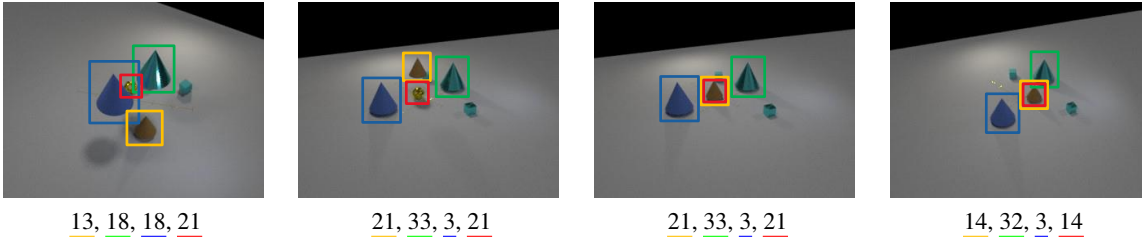


Figure 5. Example answers to the tracking surrogate task generated by our LRR model on CATER. Our LRR model is prompted with the “<track>” special token to solve the tracking surrogate task at randomly selected time-steps during training. Object tracks are over the 6×6 grid on the surface and are highlighted in color.