Ego4D Goal Step Challenge: Exploration of Various Methods for Video-Text Alignment

Zhiwen Cao
Center for Data Science
New York University
zc1592@nyu.edu

Carla Zhao
Center for Data Science
New York University
yz5996@nyu.edu

Qihang Tang
Center for Data Science
New York University
qt2087@nyu.edu

Abstract

The Ego4D Goal Step Challenge benchmarks goal-directed behavior in egocentric video data, focusing on temporal grounding of natural language queries (NLQs) in untrimmed video streams, with a key challenge of handling multiple, non-contiguous intervals for the same query. We establish baselines using NLQ models like VSLNet and ReLER but find them limited by the task's complexity. To improve performance, we employ LaViLa for multimodal feature extraction and fine-tune it with contrastive learning, ensuring better discrimination between aligned and unaligned video-text pairs. Additionally, we introduce a DFS-based alignment method that explores temporal alignments while respecting action sequences. Experiments show our approach significantly outperforms baselines on validation datasets, achieving higher mean Intersection over Union (mIoU) scores. Future work will focus on optimizing search efficiency and extending applicability to more complicated datasets.

1 Introduction

Understanding goal-directed behavior in egocentric video data is a fundamental challenge for artificial intelligence on vision-related abilities, with applications ranging from assistive technologies to robotics and augmented reality. The Ego4D dataset[1] offers a rich and diverse collection of first-person videos, presenting opportunities for tackling tasks like temporal grounding of queries, which involves identifying the start and end times of events based on natural language queries (NLQs) in untrimmed videos. However, while temporal grounding has been extensively studied in NLQ tasks, the GoalStep task[2] introduces unique challenges that make it even more demanding.

The main difficulty lies in the inherent ambiguity of the GoalStep task. In standard NLQ tasks, each query is associated with a unique time interval within a video. This deterministic nature simplifies the modeling process, as a query corresponds to only one possible temporal grounding. In contrast, the GoalStep task permits multiple time intervals corresponding to the same query. For example, in a task such as "open the fridge," the same query might appear multiple times at different stages of a video. This ambiguity introduces significant complexity, as conventional NLQ models tend to produce identical results for repeated queries, failing to account for the diversity of possible temporal groundings.

Our approach begins by leveraging NLQ models such as VSLNet[3] and ReLER[4] as baselines. Instead of relying on the pre-extracted video features provided in the Ego4D dataset, we utilize LaViLa[5] for feature extraction. This decision is motivated by the superior representational capacity of LaViLa, which captures richer multimodal interactions between video and text, thus improving downstream temporal grounding performance. By directly extracting features using LaViLa, we gain greater flexibility and fidelity in aligning textual queries with video content, which not only provides better representation of video features but is also critical to the success of our Depth First Search (DFS) based alignment approach.

To address the unique challenges of the GoalStep task, we first adopt the Bayesian method proposed in [6] as a baseline. This probabilistic approach provides a structured mechanism for handling uncertainty in temporal grounding tasks. Then we introduce our DFS approach together with fine-tuned LaVila model. LaVila model is fine-tuned to differential video frames and queries that align and that do not align. The DFS approach is then designed to utilize the frame and query embeddings from the fine-tuned LaVila model. It can address the ambiguity inherent in Goal Step challenge by modeling multiple possible temporal groundings for each query, together with the consideration of the order and number of repeats of action queries.

2 Related Work

Egocentric Video Understanding Egocentric video understanding has emerged as a critical area in computer vision, driven by the availability of large-scale egocentric datasets[1][7][8][9]. These datasets provide extensive annotations and cover real-world, first-person perspectives, where the camera is worn by the individual. Such perspectives offer unique insights into human actions, interactions, and environments but introduce several challenges, including dynamic camera movements, occlusions, and ambiguous visual cues. These factors make egocentric video analysis significantly more complex compared to traditional third-person video understanding. Research in egocentric vision spans several tasks, including temporal action segmentation[10], activity recognition[11], object interaction modeling[12], and visual query localization[13]. Temporal action segmentation, for instance, requires dividing untrimmed egocentric videos into segments corresponding to meaningful actions, which can vary in duration and occur in unconstrained environments. Similarly, activity recognition focuses on identifying and classifying the actions from the egocentric perspective. Object interaction modeling in egocentric vision aims to understand how the camera wearer interacts with objects in their environment, which is crucial for many applications in augmented reality and human-computer interaction.

Natural Language Query Challenges The Natural Language Query (NLQ) task, a key challenge in egocentric video understanding, involves localizing a temporal window in an egocentric video that provides an answer to a text query. One of the primary challenges in NLQ is handling long-term temporal dependencies in lengthy egocentric videos. Traditional methods often struggle with capturing these dependencies, leading to suboptimal performance. To address this, researchers have explored various approaches. Based on a standard span-based QA framework[14], VSLNet employs query-guided highlighting to search for matching video span within a highlighted region[3]. ReLER proposes a multi-scale cross-modal transformer with a video frame-level contrastive loss to uncover the correlation between language queries and video clips, along with two data augmentation strategies to enhance training sample diversity[4].

Multimodal Representation Learning Recent advances in multimodal learning have significantly impacted egocentric video understanding. The aim is to create unified embeddings that capture information from different modalities, such as vision and language. Researchers are increasingly leveraging multiple modalities to enhance model performance and robustness. For example, LaViLa [5](Language-model augmented Video-Language Pre-training) leverages pre-trained Large Language Models (LLMs) to create visually-conditioned narrators for dense video annotation. This approach offers advantages such as dense coverage of long videos and better temporal synchronization of visual information and text, which are particularly beneficial for understanding complex, multi-step procedures in egocentric videos. Similarly, CLIP [15] (Contrastive Language-Image Pre-training) aligns image and text features into a shared space, which has shown promise in zero-shot and few-shot learning scenarios for action recognition in egocentric videos. These multimodal approaches are being adapted for step localization and procedural understanding, enabling more nuanced interpretation of egocentric video content.

Adapters Adapters have emerged as a parameter-efficient solution for fine-tuning large pretrained models, first applied in natural language processing (NLP) as small neural modules inserted within transformer layers to allow task-specific learning while keeping the base model frozen [16]. This method significantly reduces memory usage and computational costs compared to full fine-tuning. This concept is extend by AdapterHub, enabling efficiency transfer learning accross various NLP tasks [17]. Adapters are also applied for multimodal video-text alignment, demonstrating effective adaptation with minimal additional parameters [18].

Contrastive Learning Contrastive learning has become a foundational method for representation learning, especially in tasks involving sequence alignment. Chen et al. (2020) formalized the framework with SimCLR, enabling the learning of discriminative representations by contrasting positive and negative pairs [19]. Radford et al. (2021) extended this to multimodal tasks with CLIP, aligning visual and textual embeddings through contrastive loss [20]. This framework has also been applied to sequential data, as in Xie et al. (2020), who used contrastive learning for sequential recommendation [21]. By encouraging the model to distinguish between related and unrelated queries and video frames, contrastive learning aligns naturally with Ego4D Goal Step Challenge.

Depth First Search Depth-First Search (DFS) is a classical search algorithm, whose idea is commonly used in sequence alignment tasks due to its systematic exploration of possible paths. Needleman and Wunsch (1970) has used a similar search-based approach for aligning protein sequences [22]. The book by Durbin et al. (1998) covers the application of Hidden Markov Models (HMMs) in biological sequence analysis, which used Viterbi-like path search algorithms similar to DFS for sequence prediction and alignment [23]. Given a search space of different path for sequential alignment data, DFS is a very useful technique to fully exploit and identify the optimal alignment.

3 Approach

Our approach to the Goal Step challenge consists of two main components: baseline methods and a novel DFS + Memoization algorithm. We first present our baseline approaches, which build upon established models and incorporate refinement techniques. Then, we introduce our proposed DP method, which aims to improve step localization.

Feature Extraction For both VSLNet and ReLER, we utilize features extracted using the LaViLa [5] framework. LaViLa is pretrained on 4 million video-text pairs from Ego4D, and because our task involves videos from the Ego4D GoalStep dataset, we believe it can generate accurate and contextually relevant representations for these videos. Additionally, its dual-encoder architecture enables the projection of video feature embeddings and text query embeddings into the same dimensional space, which facilitates the application of our DP approach.

Formally, let $V = \{V_i\}_{i=1}^{N}$ represent the set of videos, where each video V_i comprises n_i frames, and each video is associated with a set of text queries $T_i = \{t_i^j\}_{j=1}^{J_i}$. For every 2-second segment of video V_i, we sample 4 evenly spaced frames. These frames are processed through LaViLa's visual encoder to generate a single video feature vector for the segment. The resulting video features are denoted as $F_{v,i} = \{f_{v,i}^k\}_{k=1}^{K_i}$, where $f_{v,i}^k \in \mathbb{R}^{256}$. Similarly, the text queries are fed into the model, producing text features $F_{t,i} = \{f_{t,i}^j\}_{j=1}^{J_i}$, where $f_{t,i}^j \in \mathbb{R}^{256}$.

3.1 Baseline

Bayesian VSLNet In addition to the framework and loss functions proposed in VSLNet (details provided in the Appendix), we adopt the test-time refinement strategy based on Bayesian rules proposed by [6]. This method first groups identical text queries within a video V_i and creates an event vector from the ground truth, $\mathbf{p}_{ij} = \{p_{ij}^s\}_{s=1}^{S_i} \in \{0,1\}^{S_i}$, which indicates the occurrence of the text query t_i^j across all segments in video V_i.

To complement the conditioned span predictor introduced in VSLNet, we propose an additional predictor based on method introdued in [6]. The aggregated features, derived from the query-aware video features and video-aware query features in the VSLNet framework, are passed through the same feature encoder used in the conditioned span predictor. The refined features are then concatenated with the original input and fed into BiLSTM[24] with 3 layers. The output of the BiLSTM is projected to a linear layer with dimensions equal to $D = \max(\{S_i\}_{i=1}^{N})$, followed by a sigmoid activation function to produce $\hat{\mathbf{p}}_{ij} \in [0,1]^D$, the probability of each video segment being associated with the text query t_i^j. After masking to ignore irrelevant element in $\hat{\mathbf{p}}_{ij}$, a binary cross-entropy (BCE) loss is applied to these predictions and the event vector created earlier. The final loss function for training the model is given by:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} + \mathcal{L}_{\text{BCE}}$$

where $\mathcal{L}_{\text{span}}$ and $\mathcal{L}_{\text{QGH}}$ are the span prediction and Query-Guided Highlighting losses from VSLNet, and $\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy loss for segment-level predictions.

During inference stage, we also adopt the refinement strategy[6] to identify the most likely start and end segments associated with a given query. This method leverages the predicted probabilities $\hat{\mathbf{p}}_{ij}$, combined with temporal priors. The algorithm begins by applying a mask to filter out invalid segments based on S_i, ensuring that only relevant segments of length S_i are considered. For simplicity, only valid dimension S_i is used in following notations. It then computes a Gaussian-based prior $\mathbf{q}_{ij} = \{q_{ij}^s\}_{s=1}^{S_i}$, defined as:

$$q_{ij}^s := \mathcal{N}(s; \frac{j \cdot S_i}{m_j}, S_i \cdot \beta)$$

where j represent the text query index, m_j represents the number of occurrence of text query t_i^j, and β controls the prior shape distribution. Thus the modified prediction is defined as

$$\hat{\mathbf{p}}_{ij}^{\text{mod}} = \frac{\hat{\mathbf{p}}_{ij} \cdot \mathbf{q}_{ij}}{\max(\mathbf{q}_{ij})}$$

To rank the segments, the algorithm selects the top N segments with the highest refined probabilities, with indices determined as:

$$\text{TopIndices} = \arg\ \text{top}_N(\mathbf{p}_{ij}^{\text{mod}}).$$

For each selected segment, the start and end boundaries are determined through an iterative expansion process. Starting from the top-ranked index $k^* \in$ TopIndices, the start boundary s^* is expanded backward until $\mathbf{p}_{ij}^{\text{mod}}[s^* - 1] < \tau$, and the end boundary e^* is expanded forward until $\mathbf{p}_{ij}^{\text{mod}}[e^* + 1] < \tau$, where τ is a threshold computed as the α-quantile of $\mathbf{p}_{ij}^{\text{mod}}$ within the valid segment range, $[0, S_i - 1]$. Mathematically:

$$s^* = \max\{i \in [0, k^*] : \forall j \in [i, k^*], \mathbf{p}_{ij}^{\text{mod}} \geq \tau\}$$
$$e^* = \min\{i \in [k^*, S_i - 1] : \forall j \in [k^*, i], \mathbf{p}_{ij}^{\text{mod}} \geq \tau\}$$

The final output consists of the start and end indices (s^*, e^*) for the selected segments.

Bayesian ReLER The Bayesian ReLER framework builds upon the same architecture and training paradigm as Bayesian VSLNet, with the only modification occurring in the inference stage to accommodate the multi-scale mechanism used in ReLER. The model is trained with following objective:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} + \mathcal{L}_{\text{NPM}} + \mathcal{L}_{\text{saliency}} + \mathcal{L}_{\text{FLCL}} + \mathcal{L}_{\text{BCE}}$$

Unlike VSLNet, ReLER produces multiple probability vectors $\{\hat{\mathbf{p}}_{ij}^m\}_{m=1}^M$, each corresponding to a different scale. To make the inference suitable for the previously described algorithm, the following changes are introduced:

- Each probability vector $\hat{\mathbf{p}}_{ij}^m$ is refined using the temporal prior $\mathbf{q}_{ij}$ and corresponding mask to compute modified probabilities $\hat{\mathbf{p}}_{ij}^{\text{mod},m}$.
- These modified probabilities are stacked along the scale dimension, and a combined probability vector $\hat{\mathbf{p}}_{ij}^{\text{mod}}$ is obtained by taking the element-wise maximum across all scales:

$$\hat{\mathbf{p}}_{ij}^{\text{mod}}[k] = \max_m \hat{\mathbf{p}}_{ij}^{\text{mod},m}[k], \quad k \in [0, S_i - 1].$$

- The combined probability vector $\hat{\mathbf{p}}_{ij}^{\text{mod}}$ is then used for ranking and boundary determination as described in Bayesian VSLNet.

This modification ensures that the algorithm effectively integrates predictions across multiple scales while maintaining compatibility with the original refinement process.

3.2 Our Approach

Finetune LaVila with Contrastive Learning LaVila was originally trained on Ego4D data, originally as dual-econder and as narration generator for learning video representations from LLMs, where LLMs are repurposed to be visually conditioned "Narrators" [5]. It is very suitable for generating embeddings for video frames and action descriptions that are aligned. However, unlike other Ego4D challenges where video frames and texts (queries, narrations, etc.) are matched one-on-one, in Goal Step challenge an action can happen multiple times at different time in a video, corresponding to multiple chunks of video frames. In this case, it is very important that the model is able to generate video and texts embeddings that have high similarity for aligned pairs and low similarity for unaligned pairs. To achieve this goal, we fined-tuned LaVila with contrastive learning, and specifically, InfoNCE loss [25]:

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

Given samples X, positive sample x_{t+k} corresponding to target c_t, all samples $x_j \in X$ including both the positive sample and other negative samples, and the scoring function f_k, InfoNCE loss measures the negative value of the probability that the postive sample is correctly associated with the target, given all other candidates. In our case, the formula can be transformed to:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\text{sim}(\mathbf{q}_i, \mathbf{k}_i)/\tau)}{\sum_{j=1}^{M} \exp(\text{sim}(\mathbf{q}_i, \mathbf{k}_j)/\tau)} \tag{1}$$

By minimizing the InfoNCE loss, we are fine-tuning the ability of LaVila to align the correct pairs of video frames and action descriptions. In implementation, we sampled one positive and one negative pair for one batch of data, and we trained the model with relatively large batch size (256). Furthermore, to generalize the model's ability on various videos and actions, as well as enabling it to be able to differentiate actions that are close with no or less transition in between, the negative pairs are sampled both within same videos and a cross different videos.

Also, considering the time and cost of fully fine-tuning the LaVila dual encoder, which uses Space-TimeTransformer for the vision model and Distil BERT for text model, residual bottleneck adapters are used for efficient task-specific fine-tuning. Specifically, adapters are added after each of the last 4 layers of the vision model and the text model. The original parameters of LaVila model are frozen, and fine-tuning are performed on the parameters of the adapters.

DFS-based Alignment We aligned the video frames and action descriptions using DFS + memoization, constrained by the partial order of action sequences. More specifically, the order of the first occurrence and the number of repetitions of each action, if applicable, were provided to us. Once an action occurred, it could be repeated anytime from now on and only the number of repetitions would be recorded in the constraints. At each video frame, three possible paths could take place: 1. Stay at the current action, 2. Move to the next action, 3. Jump to a repeated action, if that action had already occurred before. The algorithm was implemented upon the similarity matrix of the finetuned LaviLa embeddings of video frames and the action embeddings, and the ultimate score of a path was computed by adding up the logarithm of the similarity scores along that path. We backtracked to find the alignment with the highest score.

4 Result

Using validation and test data, we evaluate different approaches based on the mIoU metric for the top-1 predictions at thresholds of 0.3 and 0.5. Specifically, we assess the proportion of actions with top-1 predictions that meet the IoU thresholds of 0.3 and 0.5. In Table 1, we use the same experimental configuration for all models, utilizing both our video features extracted with LaViLa and pre-extracted video features provided. The results demonstrate that our features consistently produce higher scores across all models. For all baseline models, Bayesian ReLER attains the highest score on the test data. This outcome is expected due to its fine-grained architecture and the incorporation of temporal order during the inference stage, which prevents duplicate predictions for repeated text. For our approach, it reaches the highest score on validation data. However, due to the natural of DFS to fully explore all possible alignment and find the best alignment that follows the constraints, our

Table 1: Result table

	LaViLa feature (ours)				Omnivore feature (provided)			
	Val mIoU		Test mIoU		Val mIoU		Test mIoU	
Model Name	0.3	0.5	0.3	0.5	0.3	0.5	0.3	0.5
VSLNet	19.75	14.60	24.51	17.13	12.71	8.98	17.67	11.08
Bayesian VSLNet	18.11	11.68	26.73	17.02	12.82	8.32	18.79	11.59
ReLER	21.32	16.01	26.95	19.13	**15.53**	**11.02**	21.53	14.17
Bayesian ReLER	18.91	12.07	**30.05**	**19.51**	14.06	9.19	**23.32**	**14.76**
Our Approach	**31.17**	**24.74**	-	-	-	-	-	-

approach takes significantly longer time to make predictions, especially on long videos that have too many repeated actions, even after we have ignored the existence of gaps between actions. As the test data contain a large proportion of videos that are longer and have more repeated actions comparing to the validation data, we were not able to finish evaluation of our approach on the test data within a reasonable amount of time.

5 Limitations and Future Work

While our current approach has relatively promising performance on the Ego4D Goal Step challenge, there are some limitations, as well as possible future works to extend. Firstly, our approach relies on the dual-encoder of LaVila, which is already pre-trained on other data and challenges under Ego4D. The performance of our approach on other types of egocentric videos and even none-egocentric videos needs to be further tested. Secondly, Goal Step challenge and our approach both assumes off-line scenario, where the alignment task will perform on complete videos and complete list of action descriptions. For real-world applications, such as on portable AI devices with cameras, real-time task completion should be required. Thirdly, due to the more complicated nature of the test data comparing to the validation data, our approach is only fully tested on validation data. One possible optimization is to incorporate a early stop condition based on a moving window of the a specified number of steps, but the threshold should be carefully adjusted to ensure effective pruning. Fourthly, we have ignored the existence of gaps between actions in our DFS alignment. In actual data, some videos will have a significant proportion of gaps. This limits the ability of our approach of make better alignment, as it will have to fill the gaps with other meaningful actions. Incorporating gaps in to our DFS-based alignment should be worked on in the future to further optimize its ability to make better alignment.

A Appendix - Base Model

A.1 VSLNet

VSLNet uses a sparse sampling technique to compress video features $F_{v,i}$ into S_i segments. After projecting both the video features $F_{v,i}$ and text query features $F_{t,i}$ onto the dimensional space, they are processed through a shared feature encoder. This encoder, derived from a simplified version of the embedding encoder layer from QANet [26], captures contextual information and refines the feature representations. VSLNet then applies context-query attention to compute the similarities between the video and query features, generating context-to-query and query-to-context attention weights. These weights are used to create query-aware video features and video-aware query features. The aggregated features are then passed through a conditioned span predictor to estimate the start and end probabilities for each video segment. The final predicted time interval for the query is determined by maximizing the joint probability of the start and end boundaries. Additionally, VSLNet introduces a Query-Guided Highlighting (QGH) strategy to extend the boundaries of the target moment, helping the model focus on the most relevant video regions. The model is trained with following objective:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}}$$

A.2 ReLER

Building upon VSLNet, ReLER introduces a multi-scale mechanism in the cross-modal transformer. This mechanism uses a split-and-concatenate strategy [27], where each video is divided into segments, similar to VSLNet, and each segment is processed separately to extract features. These features are refined using the Nil Prediction Module (NPM) [27], which estimates the relevance of each segment to the query. The refined features are then combined into a final representation and passed through the conditioned span predictor and QGH module used in VSLNet to predict the start and end probabilities for each video segment. Additionally, ReLER incorporates a saliency predictor, inspired by Moment DETR [28], and introduces a video frame-level contrastive loss[4]. This loss emphasizes that the similarity between text features and video features belonging to the same moment span should be higher than the similarity between text features and video frame features that fall outside of the moment span. The model is trained with following objective:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} + \mathcal{L}_{\text{NPM}} + \mathcal{L}_{\text{saliency}} + \mathcal{L}_{\text{FLCL}}$$

References

[1] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2022.

[2] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 38863–38886. Curran Associates, Inc., 2023.

[3] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization, 2020.

[4] Naiyuan Liu, Xiaohan Wang, Xiaobo Li, Yi Yang, and Yueting Zhuang. Reler@zju-alibaba submission to the ego4d natural language queries challenge 2022, 2022.

[5] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models, 2022.

[6] Carlos Plou, Lorenzo Mur-Labadia, Ruben Martinez-Cantin, and Ana C. Murillo. Carlor @ ego4d step grounding challenge: Bayesian temporal-order priors for test time refinement, 2024.

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines, 2020.

[8] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos, 2018.

[9] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288, 2011.

[10] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, 2017.

[11] Lei Wang and Piotr Koniusz. Self-supervising action recognition by statistical moment and subspace descriptors. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 4324–4333. ACM, October 2021.

[12] Yuhang Yang, Wei Zhai, Chengfeng Wang, Chengjun Yu, Yang Cao, and Zheng-Jun Zha. Egochoir: Capturing 3d human-object interaction regions from egocentric views. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[13] Hanwen Jiang, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Single-stage visual query localization in egocentric videos, 2023.

[14] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2018.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[16] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[17] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

[18] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[21] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.

[22] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[23] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[26] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018.

[27] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021.

[28] Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries, 2021.