Natural Language Queries in Egocentric Videos

Matteo Destino s329337 | Alessio Ferrari s330940 | Eleonora Guarnaschelli s334094 | Pietro Rossi s330750

Abstract

Egocentric vision provides a unique perspective on human activities by capturing visual information from the user's point of view and modeling interactions with objects, environments and people. This project addresses the Natural Language Queries (NLQ) benchmark from the Ego4D dataset, which aims to identify the temporal segment in a video corresponding to a natural language query. Building on prior works, we replicate baseline models (VSLBase and VSLNet) with various combinations of textual and pre-extracted visual features. Additionally, we propose an extension based on automatically generating synthetic queries with Large Language Models (LLMs), aiming to expand the training set and improve model generalization.

1. Introduction

Comprehending and localizing events in long, untrimmed egocentric videos from natural language queries is a key challenge in video understanding. This task, known as video moment localization, involves identifying the exact temporal segment in a video that matches a given query, with applications in video question answering, episodic memory retrieval, storytelling and robotics.

Unlike conventional video analysis, moment localization is multimodal and context-dependent, requiring alignment between visual and textual representations while reasoning over temporal context. These challenges are further amplified in egocentric videos, where the first-person view introduces noise, motion blur and subtle visual cues. In this work, we address this task using the Ego4D NLQ benchmark.

To mitigate the high computational cost of end-to-end video-language training, we adopt pre-extracted video features from two encoders: OMNIVORE [3] and EgoVLP [6]. We compare two span-based retrieval architectures, VSLBase and VSLNet, combined with BERT or GloVe text encodings. Additionally, we explore a data augmentation strategy based on generating synthetic queries with LLMs to improve model generalization in low-data or zero-shot scenarios. The code of this project can be found at the following link: https://github.com/pietrorossi24/Egovision.git.

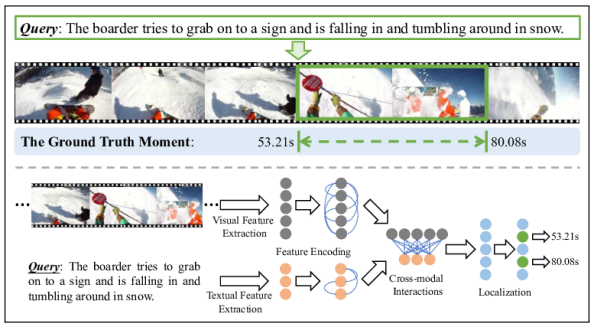


Figure 1. Natural Language Video Localization (NLVL) aims to locate a target moment from an untrimmed video that semantically corresponds to a text query (source [10]).

1.1. Episodic Memory in Ego4D

Ego4D is a large-scale egocentric video dataset comprising 3670 hours of first-person footage across hundreds of real-world environments, both indoors and outdoors, capturing a wide range of everyday scenarios such as home activities, leisure, transportation, errands, and work-related tasks, and offering a more immersive and realistic representation of human behavior compared to traditional third-person datasets.

The dataset features a diverse range of participants in terms of age, occupation, and gender, ensuring broad geographic and cultural coverage. To minimize bias from specific camera devices, Ego4D employs seven different head-mounted camera, offering varied features.
Two independent annotators provided a dense narration process, in order to organize actions and objects within the dataset. For more details refer to [4] and the official site.

These narrations are used in order to support different benchmark challenges, centered around understanding the first-person visual experience in the past (querying an episodic memory), present (analyzing hand-object manipulation, audio-visual conversation, and social interactions),

and future (forecasting activities).

For the Episodic Memory task, there are different query types; in this work, we focus on NLQ task, which is a challenging multimodal problem. It involves identifying a response track r from an egocentric video V and a natural language query Q, such that the answer to Q can be deduced from r. This task is complex due to the need for both visual recognition (e.g., events, objects, places) and linguistic reasoning (e.g., understanding relationships and reasoning).

NLQ annotations are collected by sampling 8- and 20-minute video clips. Annotators generate natural language queries to retrieve information about objects, places, and people. To minimize cognitive overload, they are given 13 query templates [Table 1] as guidance, encouraging memory-relevant queries while allowing linguistic variation through paraphrasing

Category	Template
Objects	Where is object X before / after event Y?
Objects	Where is object X?
Objects	What did I put in X?
Objects	How many X's? (quantity question)
Objects	What X did I Y?
Objects	In what location did I see object X ?
Objects	What X is Y?
Objects	State of an object
Objects	Where is my object X?
Place	Where did I put X?
People	Who did I interact with when I did activity X?
People	Who did I talk to in location X?
People	When did I interact with person with role X?

Table 1. The NLQ templates capture a diverse set of queries that humans can ask to augment their memory and recollect objects, places, and people in their everyday experience.

1.2. Dataset Exploration

The NLQ annotations cover a diverse set of scenarios (see Fig. 2), totaling 277 hours of videos and 19,2K NL queries, as summarized in Table 2 with a 60%/20%/20% train/validation/test split.

Split	Train	Val	Test
# video hours	136	45	46
# clips	*998*	*328*	*333*
# queries	*11296*	*3875*	*4005*

Table 2. **NLQ dataset statistics** across the train/val/test splits.

To explore the data, we analyzed the distribution of the most frequent words and verbs used in queries (Fig. 3), as well as the window lengths of the annotations (Fig. 4), which show a clear bias toward short windows: over 50% of queries have response windows between 0 and 5 seconds, with longer segments progressively rarer. Durations

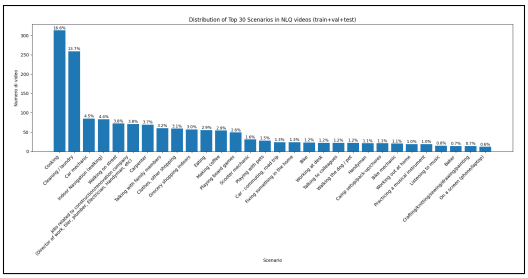
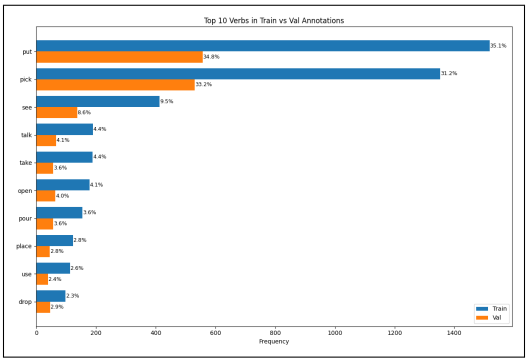
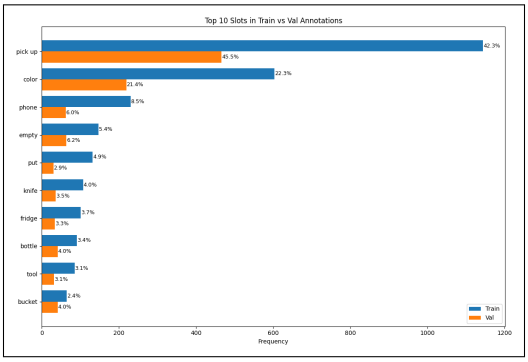


Figure 2. Distribution of queries across top 30 scenarios for the NLQ annotations, indicating a long tail over scenarios. Note that a video can contain multiple scenario labels.

exceeding one minute are grouped into a final bin. This indicates that while the benchmark is dominated by short temporal spans, it also contains a non-negligible number of longer cases.



(a) Distribution of query verbs



(b) Distribution of query words

Figure 3. Overview of verbs and words across the queries

We further analyzed the data by examining the distribution of queries across different templates and the distribution of query lengths across template, shown in Fig. 5.

Fig. 5a shows that "Where is object X before/after event Y?" is the most common template with about 2,500 queries, with a reasonable distribution over other templates. Figure 5b reveals varying response window lengths for each tem-

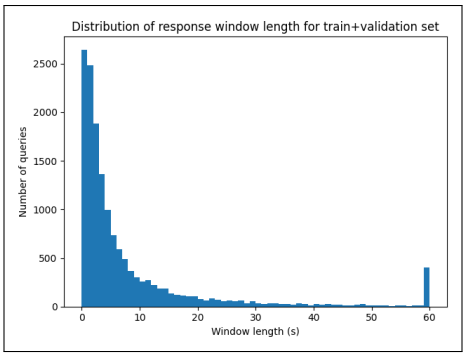


Figure 4. Distribution of response window length for NLQ. For the sake of brevity, we use the last bin to represent all windows longer than a minute.

plate. Queries involving people or quantities tend to have more variable and longer response windows, while object or location-based queries usually have shorter, more consistent durations. Outliers, often over 40 seconds, appear mainly in complex interactions or quantity-based queries. As a consequence, localization models will face greater challenges in estimating the duration of variable, long response windows, whereas for short, consistent windows even minor temporal errors (1–2 seconds) can be critical.

2. Related Work

The NLQ task has previously been investigated by using benchmark datasets, such as EPIC-Kitchens-100 [1], Charades-STA, and TACos [7]. However, EPIC-Kitchens and TACoS are limited to kitchen activities, and Charades-STA is semi-scripted and restricted to indoor scenes. Furthermore, Ego4D stands out for its scale, diversity of environments, and dense annotations.

Previous works primarily treat Natural Language Query Localization (NLVL) as a ranking task, which is solved by applying multimodal matching architecture to find the best matching video segment for a given language query. More recent studies instead regress the temporal boundaries directly or explore cross-interactions between query and video features.

A different paradigm was introduced in [9], which reformulates NLVL as a span-based question answering task: the video is treated as a context passage, the query as a question, and the target segment as an answer span. In that work, two models are proposed: VSLBase, which adapts standard span-prediction to video inputs, and VSLNet, which enhances this with Query-Guided Highlighting (QGH) to account for the continuous, long, and multimodal nature of video.

Another span-based approach is 2D Temporal Adjacent Network (2D-TAN) [8], which addresses temporal dependencies by constructing a 2D Temporal Feature Map from video features and combining it with query embeddings via a Hadamard product. A 2D convolutional network then predicts alignment scores, selecting the highest-scoring cell as the output interval.

Our work builds on the span-based QA framework: we evaluate the impact of different feature extractors on performances and compare the results with [4]. This baseline uses the SlowFast network [2], pretrained on Kinetics 400 dataset. This architecture models the distinct temporal dynamics by processing the same clip through two convolutional pathways at different frame rates: a slow pathway capturing spatial semantics via sparse sampling, and a fast pathway capturing fine-grained motion via dense sampling. Finally, the two streams are fused via lateral connections to integrate spatial and temporal information.

However, the SlowFast network does not align with our available computational resources, so we used two alternative pre-trained feature extractors, Omnivore [3] and EGOVLP [6], better explained later.

Finally, [5] introduces the Narration as Queries (NaQ) approach, which augments the dataset by converting narrations into query-video pairs, increasing linguistic diversity without additional annotations. After generating the additional NLQ samples, use these data to pre-train the VSLNet model. Then, finetune it on the original NLQ training split. Since we implemented this strategy, further details are provided later.

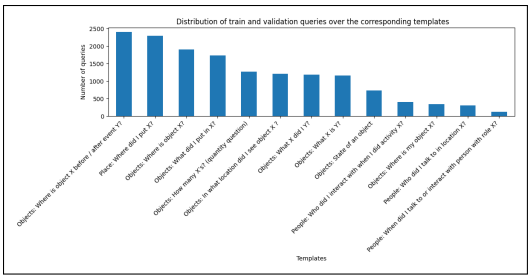
3. Methodology

In this section, we describe how to address NLVL task by adopting a span-based QA framework. We then present VSLBase and VSLNet architectures. Finally, we focus our attention to the description of the Automatic Queries Generation Extension.

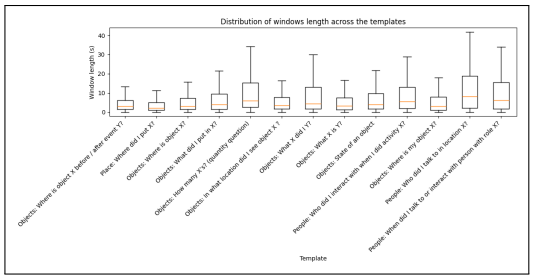
3.1. VSLBase and VSLNet Architectures

Problem Formulation. To apply span-based QA methods, each input sample is converted into a triplet $(\text{Context}, \text{Question}, \text{Answer})$, which in our case corresponds to (V, Q, A). Here, V is the sequence of visual features extracted from a single video, Q is the tokenized query embedding, and A defines the ground-truth answer span, defined as the subsequence of V delimited by start and end indices a_s and a_e, computed in the visual feature span.

VSLBase. Both V and Q are projected into a shared embedding space of dimension d, using two linear layers, resulting in $V' \in \mathbb{R}^{n \times d}$ and $Q' \in \mathbb{R}^{m \times d}$, where n is the number of extracted features and m is the number of words. These are then passed through a shared Feature Encoder, obtaining $\widetilde{V}$ and $\widetilde{Q}$. This module is composed of convolu-



(a) Distribution of queries over the corresponding templates



(b) Distribution of windows length across the templates

Figure 5. Overwiew of queries and time windows across templates

tional layers, a multi-head self-attention layer, and a feed-forward layer, with residual connections and layer normalization.

To capture cross-modal interactions between video and text, the architecture incorporates the Context-Query Attention mechanism (CQA). First, a similarity matrix $S \in \mathbb{R}^{n \times m}$ is computed between each visual feature and each word embedding. Then, context-to-query attention (A) and query-to-context attention (B) weights are computed as:

$$A = S_r \cdot \widetilde{Q} \in \mathbb{R}^{n \times d}, \quad B = S_r \cdot (S_c^\top) \cdot \widetilde{V} \in \mathbb{R}^{n \times d}.$$

where S_r and S_c are the row and column wise normalization of S by SoftMax. The final multimodal feature representation is:

$$V^q = \text{FFN}([\widetilde{V}; A; \widetilde{V} \circ A; \widetilde{V} \circ B]) \in R^{nxd}.$$

$\mathbf{V}_q$ is the used in the Conditioned Span Predictor, construct by using two unidirectional LSTMs and then two feed-forward layers, which provide the scores of start S_t^s and end S_t^e boundaries at position t.

Finally, the probability distributions of start P_s and end P_e boundaries are computed by using SoftMax, and the training objective function is defined as follows:

$$\mathcal{L}_{\text{span}} = \frac{1}{2}\left[f_{CE}(P_s, Y_s) + f_{CE}(P_e, Y_e)\right], \qquad (1)$$

where f_{CE} represents cross-entropy loss function; Y_s and Y_e are the labels for the the start (a^s) and end (a^e) boundaries, respectively.

During inference, the predicted answer span $(\hat{a}_s, \hat{a}_e)$ of a query is chosen by maximizing the joint probability of start and end boundaries by:

$$span(\hat{a}_s, \hat{a}_e) = \arg\max_{\hat{a}_s \leq \hat{a}_e} P_s(\hat{a_s}) \cdot P_e(\hat{a_e}) \text{s.t. } 0 \leq \hat{a}^s \leq \hat{a}^e \leq n$$

VSLNet. VSLBase adapts a text span-based QA framework to the NLVL task, but it does not fully capture the multimodal complexity of the problem, the continuous nature of the video input, or the need for local temporal reasoning. To overcome these limitations, VSLNet extends the original architecture by incorporating a Query-Guided Highlighting (QGH) mechanism, based on the intuition that only a subset of frames is truly informative for answering a query. QGH distinguishes relevant segments by identifying as foreground region the target moment and its surrounding context, while treating the rest as background. It operates as a binary classifier, assigning label 1 to frames within the foreground (from a_s to a_e) and 0 elsewhere. This results in an output sequence Y_h of binary values, indicating the model's confidence in each frame's relevance.

In particular, via self-attention mechanism, QGH firstly encodes word features $\tilde{Q}$ into sentence representation, which is then concatenated with each feature in V^q, obtaining $\bar{V}^q$. The highlighting score is computed as:

$$S_h = \sigma(\text{Conv1D}(\bar{V}^q)) \in R^n,$$

where σ denotes Sigmoid activation. Finally, the highlighted features are calculated by $\tilde{\mathbf{V}}_q = S_h \cdot \bar{V}^q$, and then it is used to compute the loss in 1, instead of V^q.

The model is trained end-to-end by minimizing a combined loss:

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

where $\mathcal{L}_{\text{QGH}}$ is a binary cross-entropy loss for the highlighting module.

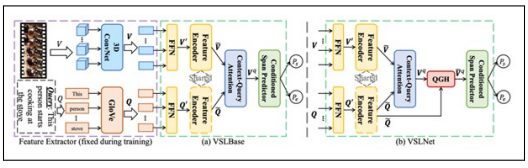


Figure 6. Architecture of the VSLBase and VSLNet models (source [9]).

Feature Extractor Both VSLBase and VSLNet process visual and textual features. For the visual modality, we rely on two sets of pre-extracted features: Omnivore and EgoVLP. Omnivore adopts a transformer-based architecture capable of handling images, videos, and 3D inputs in

a unified representation space, while EgoVLP is a dual-encoder approach pretrained on egocentric video-text pairs from the Ego4D dataset, better aligning with the characteristics of first-person video. For text encoding, we use either static GloVe embeddings or contextualized embeddings from BERT, both projected into a common hidden space before being processed by the shared feature encoder. This combination of robust video encoders and flexible text encoders enables the models to capture cross-modal interactions effectively while maintaining computational efficiency.

3.2. Automatic Queries Generation Extension

Task Presentation Despite the extensive size of the NLQ benchmark, it still covers only a small portion of the full Ego4D dataset. To fix this limitation and enrich training data for query-based video localization tasks, we propose an extension based on the Narrations-as-Queries (NaQ) strategy.

The goal of this extension is to employ a LLM, such as Gemma, to generate query-answer pairs based on randomly sampled consecutive narrations. After generating the additional NLQ samples, they will be used to pre-train the VSLNet model. The model will then be fine-tuned on the original NLQ training split, thereby enhancing its performance on real-world query generation tasks.

Query Generation For the automatic query generation, we implemented an LLM using the pre-trained causal language model "google/gemma-2b-it" from the Hugging Face Transformers library.

Gemma is a family of lightweight, state-of-the-art open models from Google, built from the same research and technology used to create the Gemini models. They are text-to-text, decoder-only large language models, available in English, with open weights, pre-trained variants, and instruction-tuned variants. Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning. Their relatively small size makes it possible to deploy them in environments with limited resources such as a laptop.

To generate queries, we first select the narrations to use as input for the model. For each video in the training set, we randomly sample two sets of three consecutive narrations to avoid bias and ensure diversity across videos. As with other large language models, a static prompt template is used to guide the model's output, then the sequence of tokens generated is decoded back into text. This approach enables the automatic generation of relevant queries from a dataset of visual narrations, offering the potential to expand the dataset to around 400k queries (one for each narration).

We analyzed generated queries and developed a function for automatic query filtering to ensure that they do not contain explicit references to the original narrations or violate the constraints specified in the prompt.

Time Window Generation from Narrations Narrations are tied to specific timestamps. To train VSLNet, we must estimate the temporal window of each described action. We adopt the *temporal response jittering* technique from [5], which converts narration timestamps into video-conditioned temporal windows.

Given a narration timestamp t_i in video V_j, we define a seed window via the contextual variable-length clip pairing strategy:

$$\bar{R}_i = \left[t_i - \frac{\beta_j}{2\alpha},\ t_i + \frac{\beta_j}{2\alpha} \right]$$

where β_j is the mean interval between consecutive narrations in V_j, and α is the mean of all β_j across videos.

Next, we apply randomized expansion and translation to obtain

$$R_i = [(\bar{t_c} - \delta_t) - s\Delta,\ (\bar{t_c} - \delta_t) + s\Delta]$$

where Δ is half the width of $\bar{R}_i$, $\bar{t_c}$ its center, $s \sim U[1, S]$ an expansion factor, and $\delta_t \sim U[-T, T]$ a translation factor. Here, δ_t models temporal uncertainty, while s expands $\bar{R}_i$ to match the distribution of response windows. S is tuned on validation, and $T = (s-1)\Delta$ ensures $\bar{R}_i \subseteq R_i$.

Finally, the generated time windows are paired with narrations and saved for model pretraining.

4. Experiments

In this section, we present the technical details and the results obtained by a comparative analysis between the implemented models and the extension.

4.1. Experimental Settings for VSL

Implementation details In our experiments, we train our model using pre-extracted video features, such as OMNIVORE and EgoVLP, along with text features from BERT and GloVe, exploring all possible combinations. To ensure a fair comparison with the benchmarks presented in [4], we adopted the same hyperparameter settings used in the baseline models. We did not perform additional hyperparameter tuning due to resource constraints, as all experiments were conducted on Google Colab.

As specified in the official documentation, we set the dimension of all hidden layers in the model to 128, and the head size of the multi-head attention mechanism is set to 8. Parameter optimization is carried out using the AdamW optimizer with an initial learning rate of 0.0001, a linear decay of the learning rate, and gradient clipping set to 1.0. To prevent overfitting, a dropout rate of 0.2 is applied.

The model is trained for 200 epochs with a batch size of 16, and we select the model with the best performance on the validation split.

Metrics We adopt "R@k IoU =m" and "mIoU" as the evaluation metrics, following [9]. The recall@k IoU=m, with k = 1, 5 and m = 0.3, 0.5 . This metric computes the percentage of times at least one of the top k predicted candidates have an Intersection-over-Union (IoU) of at least m. While "mIoU" is the average IoU over all testing samples.

4.2. Experimental Settings for Estension

Implementation details The LLM model requires different inputs to generate the additional queries:

- *prompt template*: it directs the model to generate concise, open-ended questions that are implicitly derived from the narration focusing only on observable visual elements.
- *max_new_tokens=150*: it defines the maximum number of tokens the model can generate in the output.
- *do_sample=True*: this setting enables the sampling mode, allowing the model to generate not deterministic outputs.
- *temperature=0.7*: it controls the randomness of the output. A value of 0.7 introduces some variability in the response produced.
- *top_p=0.9*: it limits the selection of possible tokens to the top 90% of probability mass, promoting more relevant and coherent query generation while maintaining diversity.

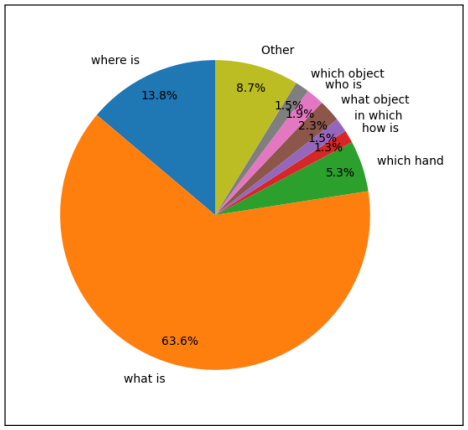
Pretraining with Narrations and Fine-tuning A key advantage of using narrations for pretraining is their large scale: the Ego queries dataset for NLQ contains nearly 400k narrations, two orders of magnitude more than the 11.3k training queries in the NLQ dataset. However, generating each query has a non-negligible computational cost. Due to resource constraints, we generated only 4440 queries, far fewer than the total narrations available.

Pretraining on this augmented dataset was performed for 20 epochs. While more epochs could improve model generalization, the limited query set required this restriction, as longer pretraining would risk overfitting. Empirical results supported this choice, with best performance achieved after just 6 epochs; the corresponding model weights were then used to initialize the fine-tuning stage.

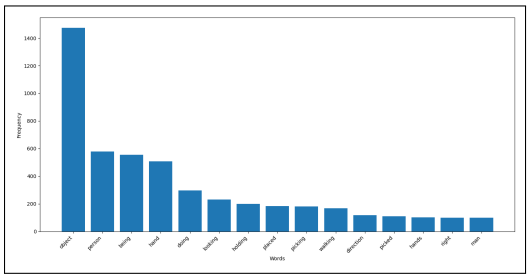
To incorporate the fine-tuning extension of this project, we adapted the authors' code by adding support for loading pre-trained weights from a specified path and ensuring their correct integration into the training procedure. Although fine-tuning is commonly performed with fewer epochs, the relatively brief pre-training phase required an alternative strategy. Therefore, we extended fine-tuning to 200 epochs to evaluate performance differences across different weight initialization, while keeping all other parameters consistent.

4.3. Generated Query Analysis

Now we analyse the quality of the query generated by the LLM. Various prompts and combinations of the previously described parameters were evaluated to identify a good trade-off between the diversity of generated queries and their consistency with the source annotations. The quality of the queries produced by the LLM was initially analysed through exploratory plots in Fig 7. Subsequently, low-quality queries were filtered out. This filtering process resulted in the retention of just over 4,000 queries from an initial set of 4,440 narrations.



(a) Beginning of the query



(b) Main words used

Figure 7. Query generation analysis

The generated queries exhibit a concentration on a limited number of primary structures (Fig. 7a), and some are too generic, referring to objects and person (Fig. 7b) in a broad sense without precise specificity regarding the actions performed. Some examples are reported in the following Table.

Category	Template
Objects	Where is the mouse?
Objects	Where is the book?
Objects	What is in the container?
Objects	What object is being removed from the plate?
Objects	What is the person holding?
Objects	What is the person looking at on the phone?
People	Who is speaking to whom?
Place	In which direction is the car driving?
Place	Where is the food pack placed on the counter?

The suboptimal quality of some generated queries is at-

tributed to two primary factors. Firstly, the performance is inherently limited by the specific LLM used; a more advanced model is expected to yield improved results. Secondly, the annotations provided to the LLM for this task were not pre-filtered, unlike those used for official query generation. Consequently, some annotations contained information that was not highly representative of the video content or described actions repeated across various time instances, making it challenging to formulate a unique query for a precise moment in time.

Despite these limitations, we opted to proceed with the pre-training phase and evaluate the obtained results.

4.4. Results

To assess model performance, we generated several plots for each model.

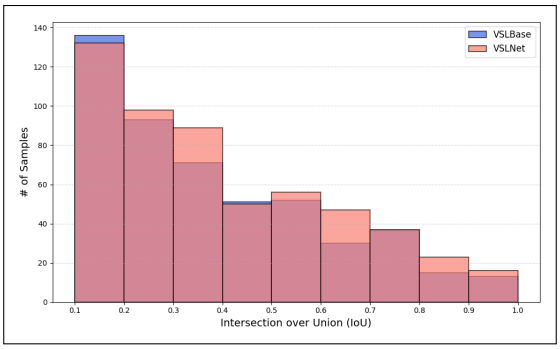
Firstly, we focus on comparing the VSLBase and VSLNet models by analyzing the IoU distribution of their predictions on the validation set, as illustrated in Fig. 8a. We selected as representative example the EGOVLP and GloVe features model configuration, given its consistent performance trends across all other tested configurations.

Going into details, most predictions produced by both VSLBase and VSLNet models fall within the [0.0–0.1) range, indicating a significant number of near-miss or incorrect predictions. To better understand this phenomenon, we analyzed the subset of queries that both models consistently failed (IoU ≤ 0.1) and investigated the associated template types, shown in 8b . We observed that the distribution of failure cases across templates is consistent with the overall distribution of query frequencies (Figure 5a). This suggests that both models fail proportionally across different query types, indicating that frequent templates are not necessarily better learned, which is a desirable property that points to a balanced generalization rather than overfitting to dominant patterns.

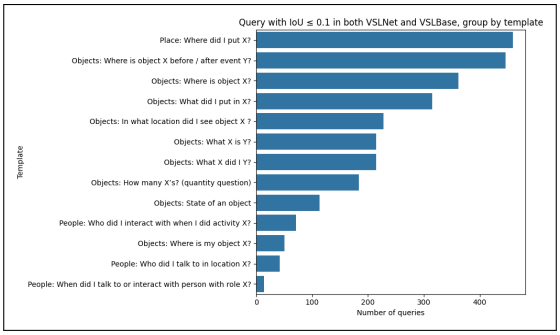
On the other hand, when comparing this trend with the distribution of average ground-truth window lengths per template (Figure 5b), it becomes evident that most failure cases are concentrated on query types characterized by particularly short annotated time intervals, which suggests the idea that both models struggle with precise localization in temporally dense or ambiguous contexts.

Finally, when focusing on higher IoU thresholds (see Fig. 8a), the VSLNet model enhanced with the Query Generation Head (QGH) consistently outperforms its VSLBase counterpart. This improvement, further supported by the quantitative metrics reported in Table 3, confirms the effectiveness of the QGH module in refining temporal localization by enabling a more accurate and robust concatenation of video-text features for the NLVL task.

We further analyzed the error patterns in the predicted moment lengths, selecting BERT as a representative case



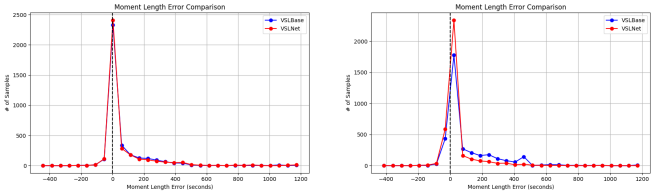
(a) Histograms of predicted results on val set under different IoUs



(b) Distribution of failure cases (IoU ≤ 0.1) over templates

Figure 8. Overview of IoU performance for EGOVLP + Glove model and failure cases analysis over templates

(with GloVe results being comparable), as shown in Fig. 9.



(a) EGOVLP + BERT (b) OMNIVORE + BERT

Figure 9. Plots of moment length errors in seconds between ground truths and results predicted by VSLBase and VSLNet, for two representative models.

The objective was to provide a cross-comparison between VSLBase and VSLNet, as well as between OMNIVORE and EGOVLP features. In these figures, we plotted the difference between the predicted moment length and the ground truth, with positive values indicating overestimation and negative values indicating underestimation. For the OMNIVORE + BERT configuration, VSLBase tends to predict longer moments, whereas VSLNet, constrained by the QGH block, tends to predict shorter moments. In contrast, for the EGOVLP + BERT configuration, this discrepancy is much less pronounced. This can be attributed to the higher

quality of EGOVLP video features: pre-trained on egocentric videos, EGOVLP seems to provides stronger temporal localization, allowing both models to predict moment durations more accurately.

For a more complete and quantitative comparison between all models, we can see the results for the evaluation metrics in Table 3.

Model	Visual Features	Textual Features	IoU=0.3 (%)		IoU=0.5 (%)	
			r@1	r@5	r@1	r@5
VSLNet	SlowFast	Bert	5.65	10.76	3.12	6.63
2D-TAN	SlowFast	Bert	5.04	12.89	2.02	5.88
VSLBase	OMNIVORE	GloVe	4.51	10.51	2.10	6.21
VSLNet	OMNIVORE	GloVe	6.09	12.44	3.34	7.54
VSLBase	EGOVLP	GloVe	7.62	15.76	4.17	9.49
VSLNet	EGOVLP	GloVe	9.01	17.46	5.07	11.08
VSLBase	OMNIVORE	Bert	4.42	10.71	2.32	6.18
VSLNet	OMNIVORE	Bert	6.74	13.01	3.54	8.27
VSLBase	EGOVLP	Bert	8.78	17.60	4.96	10.88
VSLNet	EGOVLP	Bert	10.06	18.42	5.70	12.07

Table 3. Performance on the Validation set, for NLQ task, under different IoU thresholds.

The table reports the main evaluation metrics for the models we implemented, in comparison to the baseline proposed in [4]. These quantitative results, measured by R@1 and R@5 at IoU thresholds of 0.3 and 0.5, confirm the trends observed in the previous plots. In fact, EgoVLP features significantly outperform those relying on OMNIVORE, when the model architecture remains the same. This advantage can be explained by the fact that EgoVLP, which is specifically designed for video language understanding, incorporates rich visual and linguistic representations that are tightly aligned with the nature of the Ego4D dataset. This enables the model to perform better in tasks involving the understanding of complex interactions between video and language. On the other hand, OMNIVORE, though a robust multimodal feature extractor, lacks the task-specific optimization found in EgoVLP, which likely leads to suboptimal performance in comparison.

Several other key observations emerge from these results. Models leveraging pre-trained textual features from BERT consistently outperform those relying on GloVe embeddings. This performance gap can be explained by the ability of BERT, as a transformer-based architecture, to capture contextualized word representations by considering the surrounding words in a sentence. This allows BERT to better handle complex linguistic patterns and subtle semantic relationships. In contrast, GloVe provides static word embeddings without contextual information, making it less effective in representing nuanced language and thereby limiting the model's capacity to fully interpret the textual input.

In addition, both OMNIVORE and EGOVLP provide better results with respect to the SlowFast features. In particular, the SlowFast architecture, originally designed for action recognition tasks, shows lower performance in this setting. Its separate processing of fast and slow temporal pathways may be insufficient to capture the detailed spatial and temporal alignments between video content and textual queries required for precise NLQ task.

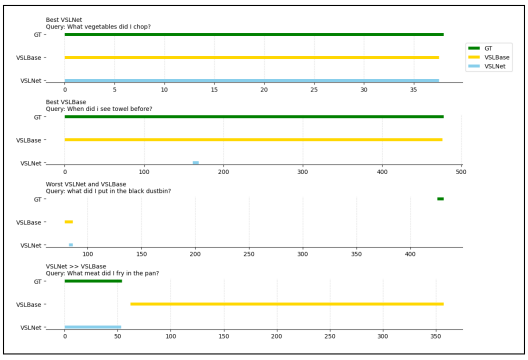
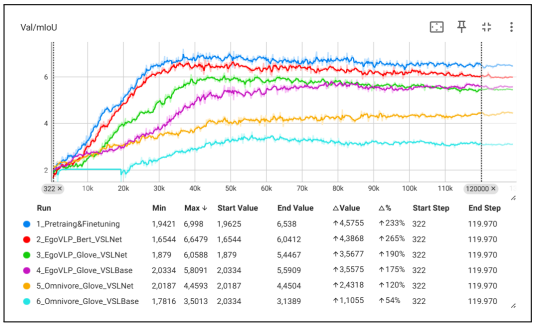
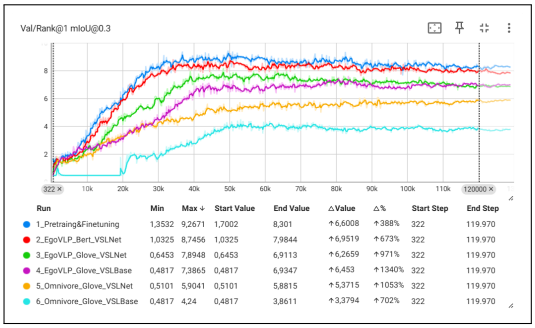


Figure 10. Qualitative comparison between VSLBase and VSLNet on four representative examples from the Ego4D NLQ validation set for EGOVLP + Bert



(a) mIoU



(b) R@1IoU@0.3

Figure 11. Plot tensorboard

Overall, the best model is EGOVLP + Bert; for this reason we provide a visual understanding of how VSLBase and VSLNet behave across different types of queries for these feature extractors (see Fig 10).

The figure shows some representative examples comparing their predictions against ground truth (GT) temporal intervals. These examples illustrate not only the rela-

tive strengths of each model but also how they can fail on the same queries, emphasizing the importance of combining quantitative metrics with qualitative analysis.

Finally, to provide a general comparison among all models and to assess the performance of the model with fine-tuning, Figure 11 shows two plots illustrating the evolution of the main metrics during training.

The general trend observed in both plots shows a similar behavior in all models. Initially, there is a clear improvement in the evaluation metrics, with some fluctuations, until a peak is reached. After this point, the metrics stabilize and converge around 50k-60k iterations. Towards the end of the training process, some models begin to decline, indicating signs of overfitting. This suggests that the models, after a certain number of iterations, start to memorize the training data rather than generalizing well to unseen data.

It is evident that the best-performing model is the one employing pre-training and fine-tuning with our proposed extension. This result underscores the effectiveness of our extension, demonstrating that even a modest increase in generated data can lead to improved performance.

4.5. Conclusions

This work provides a comparative analysis of methodologies and models for the NLVL task. Our experiments confirm that VSLNet, thanks to its query-guided highlighting strategy, outperforms VSLBase in accurately localizing NLQ segments. We further analyzed the influence of different visual and textual feature extractors, underlining the importance of their appropriate selection. While no hyperparameter tuning was performed, we believe that further optimization could lead to even better results with respect to current benchmarks.

Additionally, we proposed a simple data augmentation strategy based on converting narrations into queries using a LLM. Despite resource limitations restricting us to 4400 generated queries, this approach proved promising for enhancing performance on the Episodic Memory benchmark. Future work could explore scaling up this augmentation process to fully exploit the large pool of available narrations, potentially enabling more robust and generalizable pretraining.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. In *IJCV*, pages 33–55, 2022. 3

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 3

[3] Rohit Girdhar, Dhruv Mahajan, Alexander Kirillov, Christoph Feichtenhofer, and Kaiming He. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 1, 3

[4] Kristen Grauman, Vinay Bettadapura, Gedas Bertasius, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1, 3, 5, 8

[5] Hsin-Ying Liu, Rowan Zellers, Kristen Grauman, Devi Parikh, and Mohit Bansal. Naq: Leveraging narrations as queries to supervise episodic memory. In *CVPR*, 2024. 3, 5

[6] Rishi Rohra, Jean-Baptiste Alayrac, Antoine Miech, et al. Egocentric video-language pretraining. In *CVPR*, 2023. 1, 3

[7] Anna Rohrbach, Lisa Anne Hendricks, Marcus Rohrbach, Bernt Schiele, and Kate Saenko. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5267–5275, 2017. 3

[8] Da Zhang, Xiyang Dai, Xin Wang, Humphrey Shi, and Murray Campbell. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2020. 3

[9] Da Zhang, Xiyang Dai, Xin Wang, Yuchen Zhu, Lorenzo Torresani, Christoph Feichtenhofer, Humphrey Shi, and Murray Campbell. Span-based localizing network for natural language video localization. In *ACL*, 2020. 3, 4, 6

[10] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Zhou, and Rick Goh. Natural language video localization: A revisit in span-based question answering framework. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021. 1