

Interactive Episodic Memory with User Feedback

Supplementary Material

6. Supplementary Overview

In this supplementary material, we provide further information about dataset generation, model evaluation, and limitations of our work. All code and datasets can be found on our project page: <https://nsubed11.github.io/refocus>.

The supplementary material is sectioned as follows:

- Supplementary video showing qualitative results as discussed in Sec 4.3
- Query generation for GoalStep and HD-EPIC Datasets in 6.2
- Feedback generation for Ego4D-QnF, GoalStep-QnF, and HD-EPIC-QnF Datasets in Sec 6.3 as stated in Sec 3.2
- Evaluation Setup as stated in Sec 4.1
- Additional Model Analysis as discussed in Sec 4.3
- Commercial LVM Evaluation as stated in Sec 4.3
- Human User Feedback Collection as discussed in Sec 4.3
- Limitations and Future Work.

6.1. Supplementary Video

We provide additional qualitative results and comparisons for our approach ReFocus in video format on our project page: <https://nsubed11.github.io/refocus>. The page showcases a variety of scenarios, highlights interactions with user feedback, and includes comparisons with baseline methods. We also present failure cases where our approach struggles to effectively utilize the information provided through user feedback.

6.2. Extending GoalStep and HD-EPIC to EM-NLQ

To evaluate our approach across diverse scenarios and activity types in egocentric video, we extend two popular datasets, GoalStep [34] and HD-EPIC [29], to be compatible with the EM-NLQ task and with our feedback-based setup. While Ego4D-NLQ [7] covers a wide range of daily activities, GoalStep contains structured, multi-step tasks driven by explicit goals, and HD-EPIC focuses on kitchen-centered activities. While GoalStep and HD-EPIC provide rich annotations, they do not include EM-NLQ style natural language queries or feedback. We therefore augment these datasets with queries first, then along with Ego4D, with feedback annotations such that we enable controlled evaluation on these diverse cases for our model and the baselines. Next, we describe the query generation process for GoalStep and HD-EPIC.

For these datasets we leverage the provided text-based steps or narrations annotations. That is, we have a given set of text annotations $\{n_1, n_2, \dots, n_m\}$ and their corresponding

video spans $\{s_1, s_2, \dots, s_m\}$ sorted by their timestamps for an egocentric video \mathcal{V} . Then, for each narration n_x , we collect q textual NLQs $\mathcal{Q}_x = \{Q_{x,1}, Q_{x,2}, \dots, Q_{x,q}\}$ with response span s_x by prompting a Large Language Model (LLM), Qwen-3-8B [36], with previous, current, and next narration i.e. $\mathcal{Q}_{n_x} = \text{LLM}(n_{x-1}, n_x, n_{x+1})$. Additionally, we provide the LLM with the NLQ templates used in Ego4D-NLQ to collect annotations along with 8 in-context examples. Figure 8 shows an example showcasing the prompt used and output generated.

We refer to the query-extended version of these datasets as GoalStep-Q and HD-EPIC-Q, and Table 6 shows the statistics of these dataset along with Ego4D-NLQ across multiple splits. On average we notice that these datasets have longer video duration compared to Ego4D-NLQ, and also have very different response span durations, allowing us to test the models on diverse scenarios.

Dataset	Split	# \mathcal{Q}	# \mathcal{V}	\mathcal{V} Dur.(s)	\mathcal{R}^q Dur.(s)
Ego4D-NLQ	Train	13847	1271	529	11.3
	Val	4552	415	548	10.75
GoalStep-Q	Train	13849	606	1014	31.6
	Val	1554	35	960	30.8
	Test	3000	35	949	29.9
HD-EPIC-Q	Train	13849	17	938	2.0
	Val	1554	15	1053	2.2
	Test	3000	15	1061	2.2

Table 6. Summary of EM-NLQ datasets characteristics for all splits, reporting the total number of queries \mathcal{Q} , number of videos \mathcal{V} , average video duration, and average response span \mathcal{R} duration in seconds.

6.3. Feedback Generation for All Datasets

In this section, we add more detail for the feedback generation process using our recipe proposed in Sec 3.2. As described in Sec 3.2, for an egocentric video \mathcal{V} , NLQ \mathcal{Q} , ground-truth response span \mathcal{R}^g , we sample a reference span \mathcal{R}^f , and caption both \mathcal{R}^g , \mathcal{R}^f to get \mathcal{D}^g and \mathcal{D}^f respectively. Further, we generate explanation E^g from \mathcal{Q} and \mathcal{R}^g . Finally, we use \mathcal{D}^g , \mathcal{D}^f , E^g , and the relative temporal ordering between \mathcal{R}^g and \mathcal{R}^f to generate user feedback \mathcal{F} . Next, we provide additional details about these steps.

Reference Span Sampling. As described in Sec 3.2, we sample two different types of reference spans: query-relevant and query-irrelevant spans. We describe how we sample each of these type of reference spans below:

Query-Irrelevant Reference Spans: For a query Q with ground-truth response span \mathcal{R}^q , we sample a random span \mathcal{R}^f such that temporal intersection-over-union $\text{IoU}(\mathcal{R}^q, \mathcal{R}^f) = 0$. Random spans are sampled by sampling a center timestamp and span duration. The timestamp of the span center is sampled uniformly within the egocentric video \mathcal{V} whereas the span duration is sampled from a beta distribution fitted on min-max normalized response span durations from Ego4D-NLQ training set.

Additionally, response span \mathcal{R}'^f from other queries $Q' \neq Q$ within the same video \mathcal{V} such that $\text{IoU}(\mathcal{R}^f, \mathcal{R}'^f) = 0$ are also considered as a query-irrelevant reference span for Q .

Query-Relevant Reference Spans: We sample reference span $\hat{\mathcal{R}}^f$ from outputs of a pretrained EM-NLQ model. Specifically, we select the top-1 prediction $\hat{\mathcal{R}}_1^q$ from GroundNLQ [11] model pretrained on NaQ+NLQ [31] for queries where Recall@5, IoU=0.3 is 0.

Furthermore, we also sample reference spans that are visually similar to the ground truth span of Q but does not intersect with it. To do that, we create a set of potential reference spans using a sliding window method where \mathcal{R}'_i has the same number of d frames as \mathcal{R}^q and a stride of $\lfloor d/4 \rfloor$ resulting in $R = \{\mathcal{R}'_1, \mathcal{R}'_2, \dots, \mathcal{R}'_m\}$, where m is the total number of spans. We embed each reference span using EgoVideo’s ViT-1B [28] (averaging across clip embeddings if d is greater than video encoder’s frame size) to get $E(R) = \{f_1, f_2, \dots, f_m\}$, where f_i represents the embedding of \mathcal{R}'_i . Finally, we select \mathcal{R}'_{i^*} such that $\text{IoU}(\mathcal{R}^q, \mathcal{R}'_{i^*}) = 0$ and maximizes $\cos(f_r, f_{i^*})$, where f_r is the embedding of the ground-truth response span \mathcal{R}^q , and $\cos(\cdot, \cdot)$ denotes the cosine similarity between embeddings.

Span Captioning. For all response spans \mathcal{R}_i^q and \mathcal{R}_j^f , we leverage Qwen-2.5-VL-7B-Instruct [1] to generate \mathcal{D}_i^q and \mathcal{D}_j^f captions to describe the spans. We resize the smaller side of the video to 480px while maintaining the aspect ratio. Frames are sampled at 3 frames per second to a maximum of 96 frames. If the span is longer, we uniformly sample 96 frames instead. Figure 9 shows the prompt used and example output of captioning of response span. The generated description includes detailed information of the video content: location type, objects and their description, and the interactions.

Response Span Explanation. Similarly to span captioning, we use Qwen-2.5-VL-7B-Instruct [1] to generate explanation E_i^q from \mathcal{R}_i^q and Q_i . These explanation add specific detail about why the response span answers the given query, and ultimately is used during feedback generation to avoid answering the query directly. We resize and sample frames similarly to span captioning. Figure 10 shows the prompt used and example explanation of response span.

Dataset	Split	#F	$\#(\mathcal{R}^q, \mathcal{R}^f)$	$\#F/Q$	$\#F_{\text{rel}}$
Ego4D-QnF	Train	183490	46433	13.46	17739
	Val	22071	14122	5.0	1913
GoalStep-QnF	Train	131405	32829	9.51	17988
	Val	7335	4691	5.0	594
	Test	14083	9038	5.0	1178
HD-EPIC-QnF	Train	117016	29801	9.14	16508
	Val	7770	4973	5.0	630
	Test	21418	9623	5.0	1258

Table 7. Statistics of the feedback datasets. For each split, we report the number of feedback ($\#F$), the number of response span and reference span pairs from which feedbacks were generated as $\#(\mathcal{R}^q, \mathcal{R}^f)$, the average number of feedback per query ($\#F/Q$), and the number of feedback from query-relevant reference spans ($\#F_{\text{rel}}$).

Feedback Generation. Finally, we use the generated \mathcal{D}_i^q , \mathcal{D}_j^f , and E_i^q to generate user feedback. In addition to above generated captions, we provide the relative temporal ordering of \mathcal{R}_i^q and \mathcal{R}_j^f to a reasoning LLM, Qwen-QwQ-32B-AWQ [35], to generate the feedback $\mathcal{F}_{i,j}$. We leverage a reasoning LLM due to the complex reasoning required to generate an effective user feedback.

For feedback generation, we sample 5 in-context examples favoring examples with the same query template as the current query. We prompt the model to add more detail about the query subject, contrast the reference span, use relative temporal order, or mix these information to create a feedback. Figure 11 shows the a truncated version of an example prompt for feedback generation and its output.

Simple Temporal Feedback. We also generate synthetic temporal feedback, i.e., feedback that only contain information that guides to search before or after the current reference span. We sample temporal user feedback from pre-determined set of feedbacks that asks the model to search before or after the current reference span. This type of feedback are also generated while training as a form of data augmentation. See Sec 6.4 for more detail.

Dataset Generation and Statistics. Since Ego4D-NLQ test set is kept private, we only generate training and validation splits for Ego4D. We sample 6 random reference span per video \mathcal{V}_i in the datasets. Similarly, we sample 5000 and 1000 query-relevant spans for Ego4D-NLQ train and validation splits whereas we sample 5000, 333, and 666 query-relevant spans for train, validation, and test splits respectively for GoalStep-Q and HD-EPIC-Q. For each query in Ego4D-NLQ, we sample 3 reference span to generate feedback whereas we sample 2 for GoalStep-Q and HD-EPIC-Q.

For each query Q_i in validation and test splits, we sample 5 feedback, such that 2 feedback from query-relevant spans if they exist for query, and 2 from query-irrelevant span, and one is a simple temporal ordering feedback. Table 7 shows the statistics of the generated Ego4D-QnF, GoalStep-QnF, and HD-EPIC-QnF datasets.

Clauses Extraction from Feedback As described in Sec 3.3, we use an LLM to extract 3 different types of information (i.e., information the response span should contain, information the response span should not contain, and whether to search before or after the current reference span). We leverage Qwen3-8B with in-context examples to extract these *contains*, *not contains*, and *temporal* clauses. We provide the query Q_i and its corresponding feedback $\mathcal{F}_{i,x}$ and prompt the LLM to break the feedback down into each of the 3 different clauses if they exist in the feedback. Figure 12 shows the prompt we use and an example output.

6.4. Evaluation Setup

In this section, we provide additional details about the methods used in our experiments.

TimeChat [33]: TimeChat uses video Q-Former along with timestamp aware frame encoders to align visual context with temporal information. TimeChat finetunes LLaMA-2 [38] using LoRA [12] on the TimeIT dataset on multiple tasks that require fine-grained temporal understanding. We evaluate TimeChat without any training on user feedback. Instead, we modify the existing temporal video grounding prompt slightly to add the reference span and user feedback information similar to the Gemini-2.5-flash prompt in Figure 15.

UniTime [20]: UniTime leverages dynamic scaling of spatial features, interleaves timestamp information with frames, and hierarchical inference strategy to handle long videos. UniTime finetunes Qwen2-VL [39] using LoRA on several temporal localization datasets notably NaQ [31] and Ego4D-NLQ [7] resulting in the best performance on Ego4D-NLQ among LVLM methods. We evaluate UniTime without any further training or finetuning on user feedback. To adapt UniTime to user feedback, we use a prompt that includes the reference span information along with the user feedback similar to the Gemini-2.5-flash prompt in Figure 15. For the finetuned setting, we finetune on all 3 EM-QnF datasets simultaneously. We do so by sampling equal number of query and query with feedback samples for 1 epoch.

GroundNLQ [11] GroundNLQ uses multi-scale modules to construct the text-aware video feature pyramid and lightweight decoders to find and estimate moment boundaries. In our experiments, we use global attention instead of local attention. To adapt the base model to feedback input, we simply concatenate the text features with prediction span

and feedback features to form a new query representation. We take GroundNLQ model pretrained on NaQ+NLQ [31] dataset and finetune with a learning rate of 10^{-6} and a batch size of 16.

OSGNet [5] OSGNet is a state-of-the-art approach that integrates object features from CoDETR [48] to capture object information not represented with video features. It also uses shot level contrastive learning to understand the camera wearer’s attention from frequent movements inherent to egocentric videos. For our experiments, we take a pretrained OSGNet model pretrained on NaQ+NLQ and finetune on EM-QnF datasets. We follow the same concatenation with prediction span and feedback features to form a new query to adapt OSGNet to feedback data. We disable shot-level contrastive loss and finetune with a learning rate of 10^{-4} and a batch size of 4.

Finetuning with User Feedback. We finetune all EM-NLQ models on NLQ-only and user feedback data simultaneously. For all experiments, we randomly sample equal number of NLQ-only samples and NLQ with user feedback samples per epoch. When training on combined datasets, we sample both NLQ-only and user feedback equally from each of the 3 datasets (Ego4D-QnF, GoalStep-QnF, and HD-EPIC-QnF). So across many optimization steps, each method is trained on equal share of NLQ-only and user feedback data from each of the datasets.

We also generate simple temporal user feedback described in Sec 6.3 during finetuning. Each epoch, we generate 20% of existing feedback data as these simple temporal feedback.

6.5. Additional Model Analysis

Query Only Performance. In Table 8, we show the query-only performance of our approach compared to other baselines. For both GroundNLQ and OSGNet, we see similar performance with and without our approach ReFocus. Nonetheless, for both EM-NLQ models, our approach lead to significant improvements when feedback is available, as shown in Table 1.

Giving User Feedback on Incorrect Samples. Here, we show performance of each method where we subset QnF dataset for each method to include only those NLQs for which the method fails when provided with the NLQ only, i.e. achieves Recall@5, IoU=0.3 = 0 for the NLQ. Each method is evaluated on a its own subset to understand how well each method utilizes user feedback. Table 9 shows the performance of GroundNLQ method on these subsets.

Zero-Shot Evaluation Results. We show other recall metrics not shown in previous zero-shot evaluation (Table 2) for different methods trained only on the Ego4D-QnF

Method	Ego4D-NLQ				GoalStep-Q				HD-EPIC-Q			
	IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5	
	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5
TimeChat	1.78	N/A	0.86	N/A	1.40	N/A	0.40	N/A	0.13	N/A	0.00	N/A
UniTime	25.04	N/A	15.58	N/A	11.93	N/A	6.80	N/A	5.57	N/A	2.47	N/A
OSGNet	29.22	55.89	20.03	42.57	29.62	59.24	24.18	51.84	14.43	37.77	9.51	25.14
ReFocus(OSGNet)	29.26	54.46	20.49	42.00	29.95	58.76	24.70	51.29	14.51	37.06	9.59	24.84
GroundNLQ	29.00	54.92	21.13	42.35	23.07	52.87	17.40	43.30	11.23	32.87	6.60	20.60
ReFocus(GroundNLQ)	29.77	55.03	21.44	42.29	21.90	50.80	16.70	41.30	12.10	34.17	7.03	21.70

Table 8. Performance comparison on NLQ-only datasets Ego4D-NLQ, GoalStep-Q, and HD-EPIC-Q.

Method	Ego4D-QnF					GoalStep-QnF					HD-EPIC-QnF				
	#F	IoU = 0.3		IoU = 0.5		#F	IoU = 0.3		IoU = 0.5		#F	IoU = 0.3		IoU = 0.5	
		$\Delta R1$	$\Delta R5$	$\Delta R1$	$\Delta R5$		$\Delta R1$	$\Delta R5$	$\Delta R1$	$\Delta R5$		$\Delta R1$	$\Delta R5$		
GroundNLQ	9898	0.42	6.59	0.18	3.22	6620	0.44	6.15	0.17	3.94	10069	0.22	4.53	0.06	2.11
ReFocus(GroundNLQ)	9863	2.72	16.42	1.40	9.64	6940	3.17	18.87	1.93	13.46	9874	1.51	12.70	0.65	6.52

Table 9. Model performance delta relative to its own failure cases with query only. Each method is test with user feedback for NLQs where the method fails (achieves R5, IoU=0.3 = 0) when only the NLQ is given.

dataset but evaluated on GoalStep-QnF and HD-EPIC-QnF in Table 10. We observe that our approach ReFocus with GroundNLQ outperforms all other methods in the zero-shot setting across all the metrics and datasets. As before, we see ReFocus results in much effective use of user feedback resulting in bigger improvements in all recall metrics compared to the baseline without ReFocus even in zero-shot setting.

Feedback on Model Failure Spans We test whether our reference span sampling strategy can generalize to model errors by constructing a targeted evaluation setting where feedback is generated only for queries on which the model fails. Specifically, for ReFocus(GroundNLQ), we identify NLQs whose top-1 prediction yields $\text{IoU} < 0.3$ under the query-only setting, and generate user feedback using the recipe described in Sec. 3.2, treating the model’s top-1 prediction as the reference span. Results are shown in Table 11. Across all datasets, we observe substantial positive deltas in both R1 and R5 at IoU thresholds 0.3 and 0.5. Notably, incorporating feedback from these spans leads to larger gains (e.g., +12.77 R1@0.3 on Ego4D-NLQ and +13.57 R1@0.3 in the Goalstep-Q) compared to EM-QnF evaluation set in Table 1. These results indicate that ReFocus can generalize to leverage feedback to correct its own high-confidence mistakes.

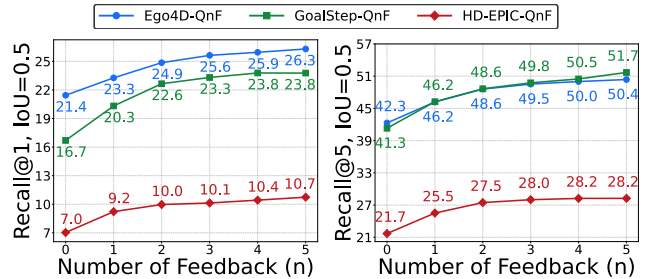


Figure 6. Multi-Turn Feedback evaluation of our ReFocus(GroundNLQ) across the three datasets at IoU=0.5.

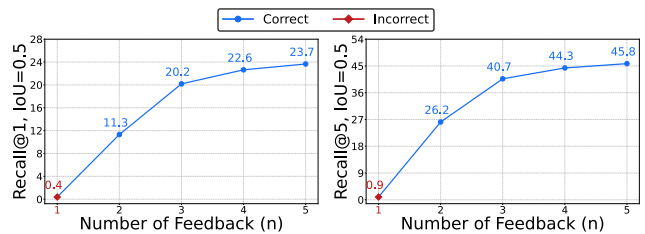


Figure 7. Noisy Multi-Turn Feedback evaluation of our ReFocus(GroundNLQ) on Ego4D-QnF at IoU=0.5. Given an initial wrong feedback (red) followed by correct ones (blue), ReFocus(GroundNLQ) is able to recover in multi-turn setting.

ReFocus with Ground-Truth FALM Ablation. Here, we extend the ablation we showed in Table 4 in Sec 4.3. Ta-

Method	GoalStep-QnF				HD-EPIC-QnF			
	IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5	
	R1	R5	R1	R5	R1	R5	R1	R5
OSGNet	14.5 ^{+0.2}	36.7 ^{+0.7}	10.0 ^{-0.3}	27.1 ^{+0.7}	5.3 ^{+0.4}	16.9 ^{+0.7}	2.4 ^{+0.3}	7.9 ^{+0.3}
ReFocus(OSGNet)	17.9^{+3.6}	42.0^{+6.8}	12.8^{+2.5}	32.4^{+5.9}	6.7^{+2.2}	18.6^{+4.5}	3.1^{+0.9}	8.6^{+2.0}
GroundNLQ	17.7 ^{+0.5}	42.2 ^{+1.6}	12.5 ^{+0.4}	31.4 ^{+1.5}	6.6 ^{+0.2}	21.3 ^{+0.3}	2.9 ^{+0.2}	10.7 ^{+0.5}
ReFocus(GroundNLQ)	20.7^{+3.7}	45.3^{+5.0}	14.5^{+2.6}	34.1^{+4.3}	8.2^{+1.6}	25.1^{+4.2}	3.8^{+1.1}	13.0^{+2.6}

Table 10. Zero-Shot evaluation of models trained on Ego4D-QnF and tested on GoalStep-QnF and HD-EPIC-QnF. Deltas (Δ) of model performance with feedback vs query-only are shown as superscripts.

Dataset	# \mathcal{F}/\mathcal{Q}	IoU = 0.3		IoU = 0.5	
		Δ R1	Δ R5	Δ R1	Δ R5
Ego4D-NLQ	12620/3196	12.77	43.7	8.11	30.14
GoalStep-Q	9355/2341	13.57	44.55	10.04	34.86
HD-EPIC-Q	10319/2634	7.61	31.99	4.24	19.61

Table 11. Performance delta for **ReFocus (GroundNLQ)** with feedback generated from its own failure spans, relative to using query only. User feedback is generated using the recipe in 3.2, where feedback is constructed from top-1 predictions for queries where the model fails (R1@IoU=0.3 = 0). # \mathcal{F}/\mathcal{Q} denotes number of feedback instances per number of incorrect queries.

Method	IoU = 0.3		IoU = 0.5	
	R1	R5	R1	R5
GroundNLQ	29.56	56.42	21.63	43.71
w. FALM	33.13	59.70	23.58	46.26
w. GT FALM _C	35.88	63.75	26.18	50.52
w. GT FALM _N	34.62	63.09	24.99	49.57
w. GT FALM _T	32.19	59.12	23.29	46.06
w. GT FALM	40.93	68.92	30.55	55.73

Table 12. Additional Ablation on ReFocus(GroundNLQ). Evaluated on a subset of Ego4D-QnF containing all types of FALM labels (see Sec 3.3).

Method	IoU = 0.3		IoU = 0.5	
	R1	R5	R1	R5
<i>Query-Irrelevant Reference Span</i>				
GroundNLQ	30.70	57.39	22.21	44.22
ReFocus(GroundNLQ)	34.09	61.03	24.42	47.32
Δ	+3.39	+3.64	+2.21	+3.10
<i>Query-Relevant Reference Span</i>				
GroundNLQ	18.29	40.77	12.44	30.00
ReFocus(GroundNLQ)	22.89	45.32	15.89	33.82
Δ	+4.60	+4.55	+3.45	+3.82

Table 13. Impact of user feedback on NLQ with different reference span relevance on Ego4D-QnF dataset.

Figure 12 shows new ablations on our proposed approach ReFocus(GroundNLQ). We look at different cases where the EM-NLQ model (GroundNLQ) is trained with the ground-truth labels for FALM instead of FALM prediction P as an upper limit performance of FALM. As before, we use labels from each of the constituent clauses extracted from the user feedback. Interestingly, compared to FALM trained on these labels, we see that using the ground truth *contains* and *not contains* labels L^c and L^k results in higher performance compared to *temporal*. We also see that we achieve similar performance with both trained and ground truth *temporal* FALM. Similarly, combining these labels results even higher performance as we observed with the trained version as well. Comparing the performance of FALM and GT FALM shows that is still room for improvement with better modeling of the feedback alignment signal, especially the *contains* and *not contains* signals.

Multi-Turn Feedback Evaluation. Figure 4 in the main manuscript shows the results of multi-turn feedback for IoU=0.3. Here, we also show the results for IoU=0.5. We show average Recall metrics after 5 different sampling of n feedback for each NLQ. Figure 6 shows the multi-feedback evaluation across the 3 datasets Ego4D-QnF, GoalStep-QnF, and HD-EPIC-QnF. We notice the same trend as in Figure 4, where the first few feedback improve recall quite significantly and eventually plateaus around 3rd or 4th user feedback. Thus, this further demonstrates the capability of our approach to handle multiple feedback without training on sequential multi-feedback data.

Effect of Noise in Multi-Turn Feedback. We also look into effect of noisy feedback on our method and test whether our method is able to recover from such feedback. We do so by initially providing an incorrect temporal feedback (i.e., pointing the model in the wrong direction) followed by normal ones. Figure 7 shows ReFocus(GroundNLQ)’s ability to recover from wrong feedback in Ego4D-QnF. We notice that at the beginning, with the initial incorrect feedback, the model only achieves <1% Recall which is expected. How-

Method	Object / place queries									People queries
	Where is X before/ after Y?	Where did I put X?	Where is X?	What did I put in X?	How many X's?	In what location did I see X?	What X did I Y?	What X is Y?	State?	Who did I interact with during Y?
OSGNet	20.5 ^{-0.3}	18.1 ^{+1.0}	15.8 ^{-0.0}	22.7 ^{+0.7}	26.4 ^{+0.2}	15.2 ^{+1.0}	28.3 ^{+1.1}	14.5 ^{+0.6}	28.5 ^{-0.8}	18.3 ^{+0.9}
ReFocus(OSGNet)	21.2 ^{-0.0}	19.5 ^{+1.0}	16.7 ^{+2.1}	25.4 ^{+3.2}	31.8 ^{+3.6}	15.8 ^{+1.8}	27.7 ^{+3.4}	19.1 ^{+1.9}	31.6 ^{+1.8}	22.6 ^{-0.0}
GroundNLQ	23.9 ^{-0.9}	21.9 ^{+0.3}	10.7 ^{+0.1}	23.8 ^{-0.2}	28.6 ^{-0.4}	9.6 ^{+0.4}	30.7 ^{+1.0}	16.3 ^{+0.1}	29.6 ^{+1.6}	19.8 ^{+0.7}
ReFocus(GroundNLQ)	25.5 ^{+0.6}	25.2 ^{+3.3}	12.2 ^{+1.1}	27.8 ^{+4.7}	31.4 ^{+2.4}	10.3 ^{+0.9}	34.3 ^{+3.4}	19.3 ^{+1.5}	33.4 ^{+3.6}	19.0 ^{-1.0}

Table 14. Evaluation with user feedback on different query types in Ego4D-QnF. Deltas (Δ) of feedback vs query-only performance are shown as superscripts.

Method	Params(B) \downarrow	TFLOPs \downarrow	Inf. Speed \uparrow
TimeChat [33]	7.97	238	0.45
UniTime [20]	8.29	142	0.07
OSGNet [5]	0.12	0.58	5.36
ReFocus(OSGNet)	0.14	0.61	4.92
GroundNLQ [11]	0.05	1.24	29.12
ReFocus(GroundNLQ)	0.08	1.64	23.52

Table 15. Efficiency comparison between different methods. Inf. Speed represents average inference speed in number of query (with/without user feedback) per seconds. We assume video features are pre-extracted for TimeChat and UniTime and both video and text features are pre-extracted for EM-NLQ methods.

ever, with subsequent feedback we see that the model is able to recover and achieves 22.6% Recall@1 performance by the 3rd correct feedback comparable to non-noisy performance.

Feedback Impact on Reference Span Types. We look at impact of user feedback that are generated from different types of reference span. As discussed in Sec 3.2 and Sec 6.3, we generate user feedback by sampling different types of reference spans. The query-relevant reference spans are spans with visually-similar features as the response spans or incorrect top-1 prediction from an EM-NLQ model. Query-irrelevant spans are either response spans from other NLQ in the same video or randomly sampled span. Table 13 shows the performance comparison between GroundNLQ and our approach ReFocus(GroundNLQ) on the Ego4D-QnF dataset for these two types of reference spans. In general, we see that the performance on examples with Query-Relevant Reference spans is lower compared to Query-Irrelevant as we sample Query-Relevant Reference spans for difficult NLQs for which pretrained EM-NLQ model fails to find the correct response span. Notably, we find that our approach helps in both types of cases, and the absolute improvement is higher compared to EM-NLQ model

without ReFocus in Query-Relevant subset than the Query-Irrelevant subset showing that our approach is more effective in using user feedback from query-relevant reference span that might include more disambiguating information compared to irrelevant ones.

Feedback Impact by NLQ Query Templates. We show the performance of our method grouped by different query types defined in Ego4D [7] in Table 14. We observe that user feedback improves most on queries that ask about an object rather than the location of an object. Interestingly, we find that ReFocus slightly degrades performance on queries that is asking about people, specifically who did the user interact with. As before, we see much bigger improvement on most query types when using ReFocus when compared to the baselines.

Efficiency Comparison. We compare the efficiency of the baselines and our approach in Table 15. As expected, LVM-based methods are inefficient especially since these method requires multiple forward passes to generate new tokens autoregressively. Notably, our approach is lightweight resulting in adding an overhead of $\sim 8\%$ for OSGNet and $\sim 20\%$ for GroundNLQ on inference speed. Since GroundNLQ uses less overall parameters, FALM adds relatively more parameters to GroundNLQ compared to OSGNet resulting in higher overhead compared to OSGNet.

Additional Qualitative Results and Failure Cases. Figure 13 shows examples where the baseline GroundNLQ model fails and our approach ReFocus(GroundNLQ) finds the correct response span. Figure 13 (a) shows an example where baseline GroundNLQ fails to reason the temporal order of the video, predicting before the reference span despite the user feedback asking the model to check after. Similarly in Figure 13 (d), GroundNLQ again fails to reason about the user feedback and ends up predicting the same

Method	Ego4D-QnF				GoalStep-QnF				HD-EPIC-QnF			
	IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5	
	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5	R1	R5
Gemini-2.5-flash	15.7 ^{+1.7}	28.7 ^{+0.7}	8.7 ^{+2.7}	14.7 ^{-1.3}	8.7 ^{+2.7}	16.0 ^{+1.0}	2.7 ^{+0.7}	5.7 ^{-1.3}	6.7 ^{+4.7}	15.7 ^{+6.7}	3.3 ^{+3.3}	8.0 ^{+4.0}
ReFocus(OSGNet)	24.0 ^{+3.0}	46.7 ^{+2.7}	13.7 ^{+2.7}	29.0 ^{+3.0}	21.7 ^{+2.7}	53.7 ^{+3.7}	13.0 ^{+3.0}	35.7 ^{+2.7}	9.7 ^{+1.7}	39.3 ^{+2.0}	6.7 ^{+1.7}	21.3 ^{+1.3}
ReFocus(GroundNLQ)	8.7 ^{+8.7}	48.0 ^{+48.0}	4.7 ^{+4.7}	29.3 ^{+29.3}	9.7 ^{+9.7}	54.7 ^{+54.7}	4.0 ^{+4.0}	38.7 ^{+38.7}	4.3 ^{+4.3}	47.0 ^{+47.0}	2.0 ^{+2.0}	22.3 ^{+22.3}

Table 16. Performance comparison between Gemini-2.5-flash and Refocus(GroundNLQ) on a small 100 NLQ subset where ReFocus(GroundNLQ) fails with query-only but improves with feedback.

Method	Ego4D-QnF				GoalStep-QnF				HD-EPIC-QnF			
	IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5		IoU = 0.3		IoU = 0.5	
	Δ R1	Δ R5	Δ R1	Δ R5	Δ R1	Δ R5	Δ R1	Δ R5	Δ R1	Δ R5	Δ R1	Δ R5
Gemini-2.5-flash	8.8	18.6	4.9	8.3	3.5	9.0	1.2	3.1	4.4	13.2	2.2	6.6
ReFocus(GroundNLQ)	8.3	47.0	4.9	25.5	9.8	51.8	3.9	34.5	2.9	45.8	2.2	22.0

Table 17. Performance comparison with feedback on subset of NLQs where Gemini-2.5-flash fails with NLQ-only.

room as the reference span despite user feedback telling the model that is the incorrect room. These demonstrate our method’s ability of effectively use user feedback.

Figure 14 shows examples where our method fails. Figure 14 (a) shows our model effectively using feedback to find the correct moment but the predicted span is not aligned well with the ground truth span. Figure 14 (b) shows a case where our method focuses too much on the user feedback asking for signs and predicts a moment that only has the sign but no fire extinguisher. Figure 14 (c) shows a case where our method fails to find the correct object and confuses tuna can for bell peppers.

6.6. Commercial LVLm Evaluation

We use Gemini-2.5-flash, to evaluate the ability of state-of-the-art commercial LVLm to integrate user feedback effectively. In Table 3, we tested Gemini on a set of queries where GroundNLQ fails completely and reported metrics on 2 of the 3 QnF datasets. Here, we provide the full table for the same evaluation in Table 16.

Additionally, we provide additional evaluation but on queries where Gemini fails. That is, we first test Gemini with a set of EM-NLQs and select those that the model fails to answer correctly in its top-5 predictions, to test the model with feedback from our datasets. The resulting set contains 204, 255, and 273 feedback for Ego4D-QnF, GoalStep-QnF, and HD-EPIC-QnF respectively out of 300 total feedback from each datasets.

Table 17 compares the performance delta of NLQ with user feedback and NLQ only for Gemini-2.5-flash, GroundNLQ, and our approach on the above described subset. We find that Gemini-2.5-flash shows similar improvement to our model in recall@1, IoU=0.3 for Ego4D-QnF

and HD-EPIC-QnF, but our approach is much better at Recall@5 and better across all metrics in GoalStep-QnF. Compared to GroundNLQ without ReFocus, we see our approach is effectively using user feedback to improve across all datasets and metrics.

Comparing the deltas from Table 3 to this new test, we see that Gemini-2.5-flash improvements in Table 3 are lower. This suggests that Gemini-2.5-flash gets confused with user feedback on cases where it can find the response span with just NLQ.

We also provide the prompts used with Gemini-2.5-flash for both NLQ-only and NLQ with user feedback cases in our experiments in Figure 15.

6.7. Human User Feedback Collection

We collect human feedback by asking 8 users to assume the role of the user asking the query in the EM-NLQ task to recall past activity. The annotator’s goal is to provide feedback to find the correct response span given the incorrect reference span along with the full video and the query. We advise the annotators to not answer the query directly by looking at the correct response span and also try to avoid giving only simple temporal user feedback like search before or after the current reference span. The annotators are given access to the entire video, reference span, and the correct span to come up with a feedback. Figure 16 showcases the interface used to collect human user feedback.

6.8. Limitations and Future Work

Collecting user feedback for the EM-NLQ task is costly. This work proposes an effective method to synthesize feedback and train models for interactive EM-NLQ, yielding improved performance. However, the approach depends

on LLMs to generate feedback, and some generated examples contain hallucinations that introduce noisy training samples. Reducing hallucinations via higher-quality LLMs, improved system design, or refined prompts, would produce cleaner training data and likely further improve performance.

Additionally, our current feedback is generated based on language descriptions of reference and ground truth spans. Incorporating visual context from the referenced video spans could provide more precise, visually grounded feedback and improve quality as well as incorporating . This would, however, increase computational cost significantly because of the larger context needed to process video input.

Furthermore, while we showed that our method can handle multiple feedback by independently modeling and combining them to form a strong baseline, we did not extensively experiment with multi-turn user feedback where subsequent user feedback could refine, correct, or condition on previous feedback. Training a specialized method to handle order-aware state and belief tracking for multiple user feedback could lead to better performance. Future work could look into such methods improving this aspect for a more robust conversational system.

User:

Help me generate natural language questions (NLQs) from narrations of daily activities. You will be given a narration of an activity for which you will generate questions for, and optionally the previous and next narrations for context. The questions should be focused on the activity or the objects involved in the narration, and sometimes use the previous and next narrations to help you generate the questions.

Here are examples of good questions:

[IN-CONTEXT EXAMPLES]

Use the following **question templates** as guidance:

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

For the given narration and context below, generate 1–3 relevant questions following the question templates and the examples above.

Narration: add oil to pasta in pot

Previous narration: seive pasta

Next narration: serve food

Assistant (LLM):

What oil did I add to the pasta in the pot?

Figure 8. Example prompt used to generate GoalStep-Q and HD-EPIC-Q datasets and output generated by Qwen-3-8B. Text after Assistant is generated output. We provide 8 in-context examples.

System:

You are assisting a user with describing videos. The user will provide the video for you to describe. Your goal is to describe the video in detail such that your description can function as a replacement for the video itself.

User:

Describe the above first-person perspective video taken by someone wearing the camera on their head. Describe in detail: the location, every objects large or small in the video and their visual attributes, every interaction that occurs, and finally a summary. Be very specific about each objects or interaction you observe. Use different sections and lists to organize the description.

Assistant (LVLM):**Location:**

The video appears to be set in a small kitchen, likely within a home. The walls are light-colored, and there's a visible electrical outlet and a digital thermostat on the wall. The countertop is cluttered with various cooking utensils, ingredients, and appliances.

Objects:

- **Cooking Area:** A stovetop with multiple burners is present. On the stovetop, there are several pots and pans containing food. One pot has a wooden spoon resting inside it, another contains a yellowish mixture, possibly noodles or rice, and a third pot seems to have a soup or stew with vegetables.
- **Countertop Items:** The countertop is crowded with items including a cutting board, a knife, a container of green vegetables, a bowl of what looks like chopped meat, a bottle of oil, a bag of flour, a jar of what might be spices, and a few other miscellaneous items.
- **Appliances:** A coffee maker is visible on the counter, along with a kettle and a blender.
- **Electrical Outlet:** A standard electrical outlet is mounted on the wall near the stove.
- **Digital Thermostat:** A digital thermostat is also mounted on the wall next to the electrical outlet.
- **Window:** There's a window with vertical blinds partially covering it, allowing some natural light into the room.
- **Trash Bin:** A small trash bin sits on the counter near the stove.

Interactions:

The person filming is actively engaged in cooking. They are seen stirring the contents of one of the pots on the stove using a spoon. Their hand movements suggest they are mixing or checking the consistency of the food.

Summary:

The video captures a moment in a small, busy kitchen where someone is preparing a meal. The scene is dynamic, with the person cooking actively stirring the food in a pot on the stove. The kitchen is compact but functional, with various cooking tools and ingredients laid out on the counter. The overall atmosphere suggests an everyday cooking scenario, possibly in preparation for a meal.

Figure 9. Example Span Captioning prompt and output generated by Qwen-2.5-VL-Instruct during user feedback generation.

System:

You are assisting a user with describing videos. The user will provide the video for you to describe. Your goal is to describe the video in detail such that your description can function as a replacement for the video itself.

User:



The above first person perspective video taken by someone wearing the camera on their head. A query is asked by the camera wearer and the given video answers the given query: "What oil did I add to the pasta in the pot?". The video always answers the query. The query is always either about location of some object or action of some person or attribute of some object. First, explain why the video answers the query.

After that:

- If the query is asking about the location of some object, describe the object and its features.
- If the query is asking about some action, describe the object manipulated by the person and the action.
- If the query is asking about some attribute of some object, describe the location of the object.
- If the query asks what object was placed in given location, describe details about the location and its surroundings.

Section your answer into "Answer Explanation" and "Relevant Query Subject Details".

Assistant (LVLM):

Answer Explanation:

The video answers the query because it shows the person adding oil to the pasta in a pot, which directly demonstrates the action of adding oil to the dish.

Relevant Query Subject Details:

The subject is the person who is cooking pasta. They are using a pot on the stove and adding oil from a bottle to the pasta inside the pot. The pot is located on one of the burners of the stove, and the person is holding the bottle of oil in their right hand while using a spoon to stir the pasta in the left hand.

Figure 10. Example Response Span Explanation prompt and output generated by Qwen-2.5-VL-Instruct during user feedback generation.

User:

We are trying to collect data for episodic memory task. The task is formulated as follows:

...

Here are the guidelines for providing feedback for this type of query:

Guidelines:

1. **Do not Answer the Query Directly.** The query is primarily asking for action of placing some object inside location X. Do not provide any information about the object that was placed in the given location.
2. **Give Feedback as if You are Asking the Query.** Feedback should be either questions or first person focused statements unless the query focuses on someone else. You should give feedback as if you are the person who is asking the query and is looking for the correct video clip.
3. **Attributes of the Given Location.** Since the query is asking for what kind of object was placed in given location X, you can provide any information about the given location, from its attribute to its surrounding objects.
4. **Contrast between the Video Clips.** Use memorable contrastive information about object or action that are present/absent in the incorrect video clip but absent/present in correct video clip or the query.
5. **Relative Time.** You can use the relative time of the video clips to provide time related feedback but do not be too specific.

...

Now, I will provide some examples of good and bad feedback for given query and the video descriptions.

[IN-CONTEXT EXAMPLES]

[RESPONSE SPAN DESCRIPTION]

[RESPONSE SPAN EXPLANATION]

[REFERENCE SPAN DESCRIPTION]

Query: What oil did I add to the pasta in the pot?

Relative Time between the two video clips:

Correct video clip occurs 0.85 minutes before incorrect video clip

Provide a short feedback similar to the previous examples.

...

Assistant (LLM):

<think>

Okay, let's tackle this problem step by step.

...

</think>

```
[{
  "feedback": "The oil was added while the pot was on the stove, not after moving to the bowl. Could you check that part?",
  "explanation": "It avoids revealing the oil type (Guideline 1), uses first-person focus (Guideline 2), contrasts the location/action (Guideline 4), and implies a time when the pot was actively on the stove (Guideline 5).",
  "guidelines_followed": "1,2,4,5"
},
...
]
```

Figure 11. Example Feedback Generation prompt and output generated by Qwen-QwQ-32B-AWQ during user feedback generation. Prompt and output are truncated for brevity.

User:

You are helping me generate clauses from a sentence given as feedback about a query action/object to be searched in a video.

The query is first searched in a video to answer it. If the answer is wrong, I provide feedback about what is wrong in the answer.

The feedbacks are created after a current incorrect answer/prediction is made find the query and feedback are supposed to be used to improve the answer.

Your task is to extract simple information from the feedback to understand it. You will be given the query and the feedback for that query. Your task is to extract clauses that either says what the answer to the query should contain, what it should not contain, and temporal information regarding where to search before or after the current prediction.

The feedback usually have 3 different types of information:

1. Some information related to the query action or object to be searched
2. Some information that should not be searched for
3. Temporal information regarding where the query action/object is located in the video timeline (either before or after or null if not specified)

Your job is to extract these 3 potential types of information from the feedback. Some feedback may not have all 3 types of information.

Here are some examples to help you understand the task:

[IN-CONTEXT EXAMPLES]

Now, please extract the clauses from the following feedback. If a type of information is not present, use null for that field. Respond only with the JSON object containing the fields "contains", "not_contains", and "temporal".

Query: What oil did I add to the pasta in the pot?

Feedback: The oil was added while the pot was on the stove, not after moving to the bowl.

Clauses:

Assistant (LLM):

```
{
  "contains": "oil added while the pot was on the stove",
  "not_contains": "moving to the bowl",
  "temporal": null
}
```

Figure 12. Example Clauses Extraction prompt and output generated by Qwen3-8B.

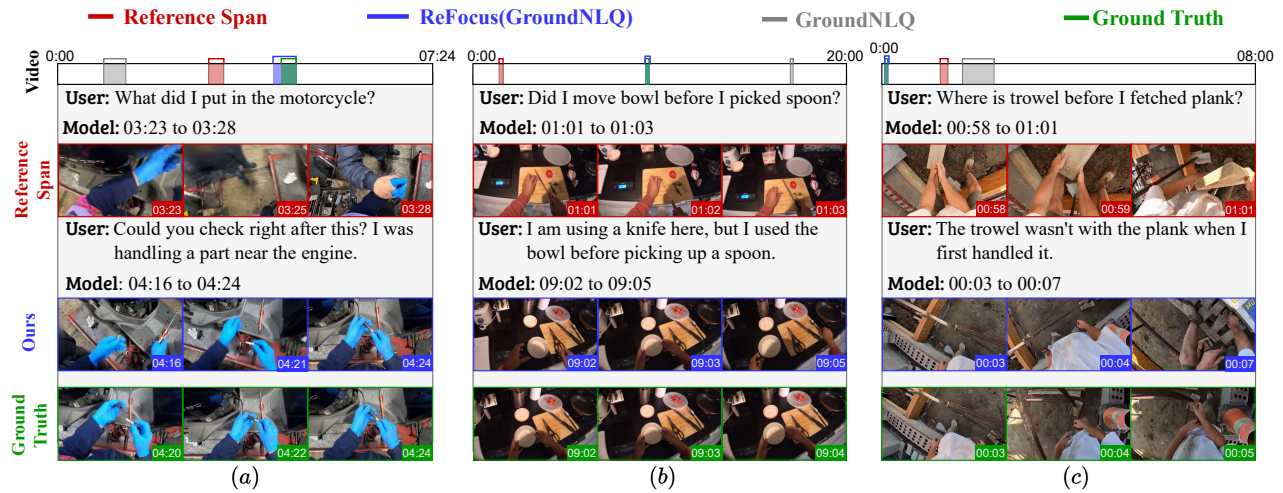


Figure 13. Additional Qualitative Results. These examples show our method improving on the baseline.

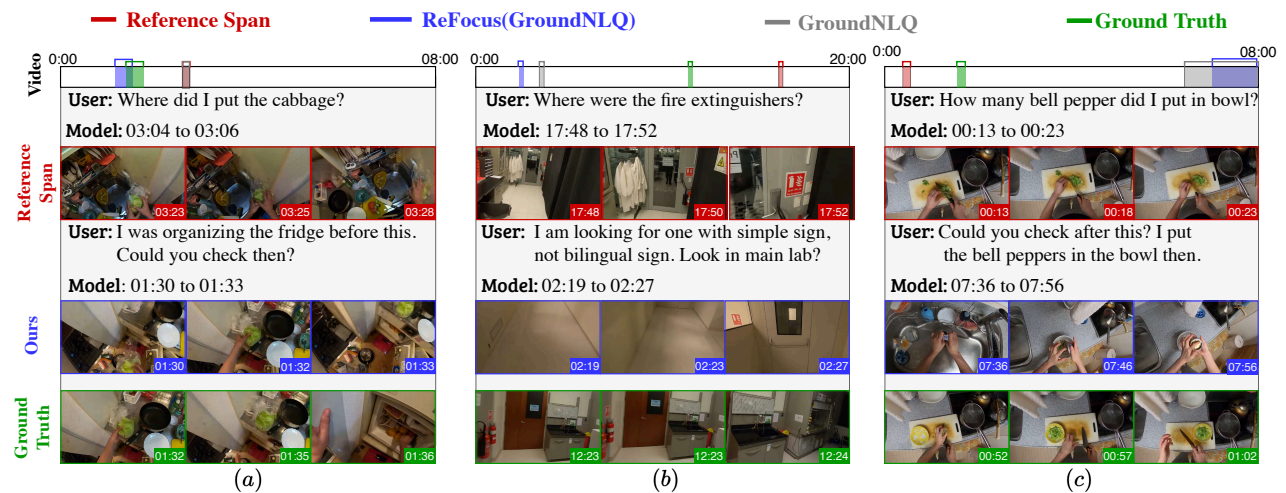
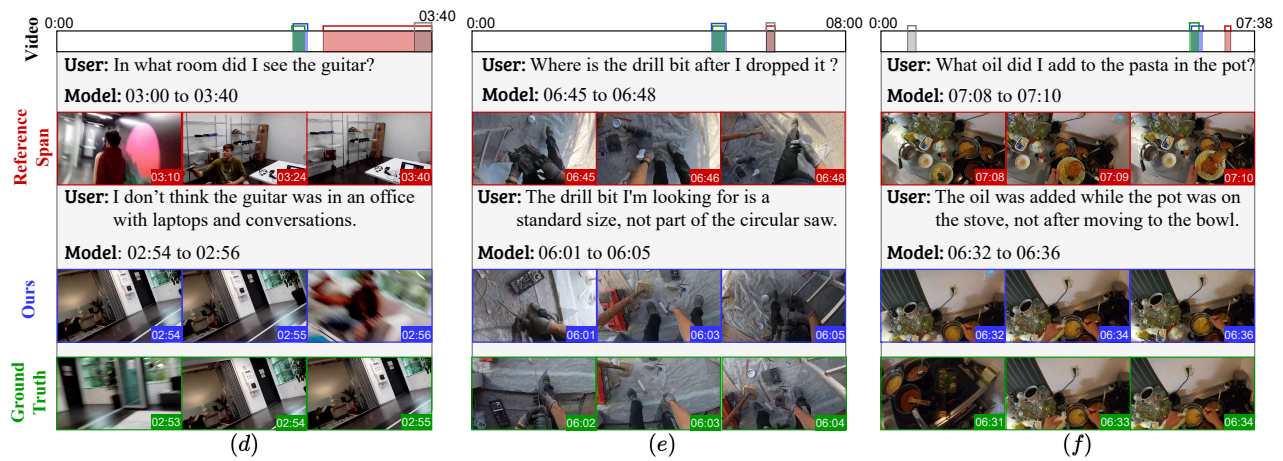


Figure 14. Additional Failure Cases. These examples show few failure cases of our method.

User:

For the given first person video, I will ask you a question to recall my past activity. Can you provide me with a moment in the video such that the answer to my question can be deduced from that moment?

Question: "What oil did I add to the pasta in the pot?"

Provide your answer in the JSON list format: [{{"start_time": "MM:SS", "end_time": "MM:SS"}}].

Answer top 5 most relevant moments in the video as a list of JSON objects sorted by relevance.

(a)

User:

For the given first person video, I will ask you a question to recall my past activity. Can you provide me with a moment in the video such that the answer to my question can be deduced from that moment? I previously asked the question and you provided a moment which is not the one that I am looking for. So, I will also give you feedback based on this previous moment to help find the correct moment. I want you to use that feedback to revise your answer.

Question: "What oil did I add to the pasta in the pot?"

Previous predicted moment was start_time: 07:08 to end_time: 07:10. This moment is not what I am looking for.

Feedback: "The oil was added while the pot was on the stove, not after moving to the bowl."

Provide your revised answer in the JSON list format: [{{"start_time": "MM:SS", "end_time": "MM:SS"}}].

Answer top 5 most relevant moments in the video as a list of JSON objects sorted by relevance.

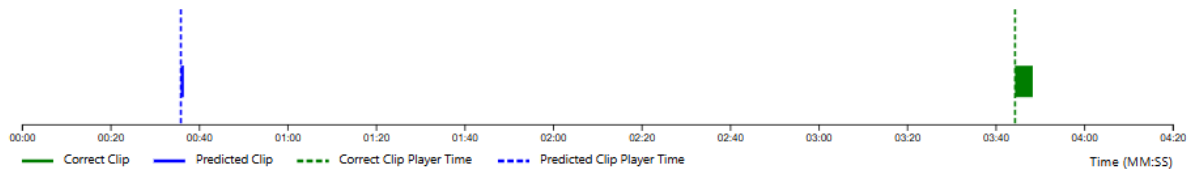
(b)

Figure 15. Example prompts used with Gemini-2.5-flash. (a) shows inference with NLQ only. (b) shows inference with NLQ and user feedback.

Full Video



Your Question: in what location did i see the piano



Predicted Clip
(This is what the model thinks you are looking for)

Play Predicted Clip (0:35 - 0:36) Fullscreen

Correct Clip
(For reference only. You don't remember this exactly)

Play Correct Clip (3:44 - 3:48) Fullscreen

▶ Play Both Together

Provide your feedback and click next to submit.

◀ Back Next (Auto Submit)

Figure 16. Human User Feedback Collection UI. Users were given a short tutorial and advised not to directly answer the question.