Object-Shot Enhanced Grounding Network for Egocentric Video

Supplementary Material

In the supplementary material, we first outlined the baseline settings on Ego4D-NLQ to establish their validity(Section A). We then conducted ablation experiments from scratch to eliminate potential bias from pretraining, and also conducted additional ablation experiments to verify the rationality of our design(Section B). Next, we categorized Ego4D-NLQ by question templates and compared model performance across these categories to the strong baseline, GroundVQA [1], highlighting our improvement in background object-related query localization (Section C). To further demonstrate the effectiveness of our model, we provided comprehensive visualizations illustrating the diverse data and our model's predictions (Section D). Finally, we introduced the model structure and other implementation details (Section E).

A. Baseline Settings

A.1. On Ego4D-NLQ v1

In Ego4D-NLQ v1, there is a significant amount of noisy data with ground truth durations of 0, resulting in predicted outputs consistently yielding an IoU of 0, making accurate localization impossible. As a result, RGNet [2] and SnAG [3] remove these noisy samples from the validation set. However, for a fair comparison across all methods, we evaluated all models on the original validation set, including the noisy samples.

RGNet. Since RGNet [2] did not release any checkpoints trained from scratch, we retrained the model (No.2 in Table 1). For the pretraining setting, we utilized the fine-tuned checkpoints published by RGNet [2] for testing on the original validation sets (No.2 in Table 2). RGNet [2] removes $N^R_{noisy} = 341$ noisy samples with ground truth durations of 0, along with $N^R_{add} = 4$ additional samples. Assuming that the predictions on these N^R_{add} samples are correct, we can adjust the evaluation result as follows:

$$\begin{cases} N^R = N_{val} - N^R_{noisy} - N^R_{add}, \\ m^R_{cor} = \frac{m^R_{ori} * N^R + N^R_{add}}{N_{val}}, \end{cases} \tag{1}$$

where N^R is the number of validation samples used in RGNet, $N_{val} = 3874$ is the number of total samples in the Ego4d-NLQ v1 validation set, m^R_{cor} represents the evaluation result after adjustment (No.3 in Table 1 and Table 2), and m^R_{ori} is the result reported in the origin paper (No.1 in Table 1 and Table 2).

Table 1. The results of RGNet trained from scratch on the validation set under different settings.

No.	Setting	R@1		R@5	
		0.3	0.5	0.3	0.5
1	Original	18.28	12.04	34.02	22.89
2	Reproduce	16.86	10.53	34.43	21.84
3	Correction	16.76	11.07	31.09	20.95

Table 2. The result of RGNet with NaQ pretraining strategy on the validation set under different settings.

No.	Setting	R@1		R@5	
		0.3	0.5	0.3	0.5
1	Original	20.63	12.47	41.67	25.08
2	Checkpoint	18.66	11.72	36.37	22.43
3	Correction	18.90	11.46	38.06	22.95

Table 3. The result of SnAG [3] on the validation set under different settings.

No.	Setting	R@1		R@5	
		0.3	0.5	0.3	0.5
1	Original	15.87	11.26	38.26	27.16
2	Correction	15.23	11.07	35.45	25.43

SnAG. Since SnAG [3] has not released complete annotations for the validation set, we used the published checkpoints to evaluate the test set (SnAG in Table 1 in the *Manuscript*) and applied a formula to adjust the results for the validation set (No.2 in Table 3). SnAG [3] removes $N^S_{noisy} = 341$ noisy samples with ground truth durations of 0, along with $N^S_{add} = 35$ additional samples. Assuming the predictions on these N^S_{add} samples are correct, we adjusted the results as follows:

$$\begin{cases} N^S = N_{val} - N^S_{noisy} - N^S_{add}, \\ m^S_{cor} = \frac{m^S_{ori} * N^S + N^S_{add}}{N_{val}}, \end{cases} \tag{2}$$

where N^S represents the number of validation samples used in SnAG, m^S_{cor} are the results after adjustment (No.2 in Table 3), and m^S_{ori} are the results reported in the origin paper (No.1 in Table 3).

B. Ablation Study

B.1. On Scratch

To further demonstrate the efficacy of our module, we conducted ablation experiments on the Ego4D-NLQ dataset

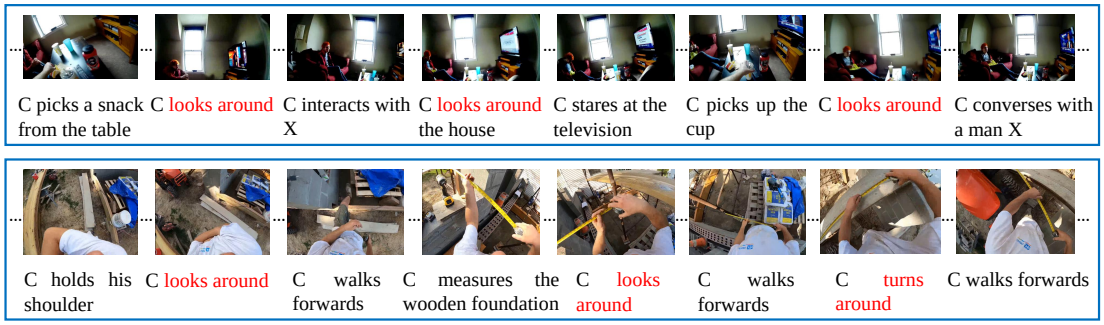

Figure 1. Illustration of captions generated by LAVILA [5] describing camera movements.

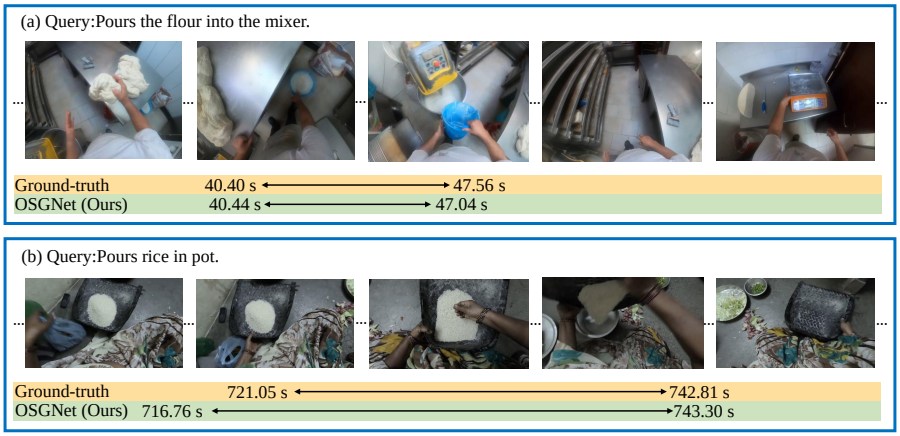

Figure 2. Illustration of grounding results on Ego4D-Goal-Step.

Table 4. Ablation studies on model structure for Ego4D-NLQ v1.

$\mathcal{L}_{con}$	Object	R@1		R@5	
		0.3	0.5	0.3	0.5
✗	✗	9.53	6.61	27.21	18.61
✓	✗	13.37	8.98	32.21	22.02
✗	✓	14.07	10.12	33.84	24.08
✓	✓	**16.13**	**11.28**	**36.78**	**25.63**

Table 5. Ablation studies on model structure for Ego4D-NLQ v2.

$\mathcal{L}_{con}$	Object	R@1		R@5	
		0.3	0.5	0.3	0.5
✗	✗	13.27	9.16	36.78	25.92
✓	✗	17.29	11.86	40.55	28.80
✗	✓	17.40	12.10	41.10	29.92
✓	✓	**18.74**	**12.72**	**41.92**	**30.34**

while excluding the NaQ pretraining strategy (NaQ [4]). The results for Ego4D-NLQ v1 and Ego4D-NLQ v2 are presented in Table 4 and Table 5, respectively.

In Table 4, removing $\mathcal{L}_{con}$ results in a 1.16% performance drop at R@1, 0.5, while excluding the object feature leads to a 2.30% decrease. Similarly, Table 5 shows that ablating the object feature and the shot branch causes declines of 0.86% and 0.62% in R@1, 0.5, respectively. These results underscore the critical role of the shot-level branch in enhancing video representation learning and highlight the importance of object-level information for the NLQ task.

Compared to the ablation experiments in the *Manuscript*, the shot branch exhibits a more significant improvement when pretraining is excluded. We attribute this to two factors. First, during pretraining, our model undergoes extensive semantic alignment training, causing the enhancements provided by the shot branch to overlap with those already acquired, thereby yielding limited additional gains. Second, because the shot branch is not included in the pretraining phase, its data alignment remains misaligned with that of the main branch, further constraining its performance improvements.

B.2. On Shot Segmentation

In Table 5 of the *Manuscript*, we validate our design by analyzing high-frequency movement-related verbs. Additionally, Table 6 presents result from segmenting videos with an average shot length of 13 seconds without using verbs. In

Table 6. Ablation studies on shot segmentation.

Method	R@1		R@5	
	0.3	0.5	0.3	0.5
Average	31.22	21.42	**58.22**	44.95
LAVILA	**31.63**	**22.03**	57.91	**45.19**

Table 7. Ablation studies on the self-mixer in the main branch.

Self-mixer	R@1		R@5	
	0.3	0.5	0.3	0.5
Self-attention	28.14	19.66	55.38	42.64
BiMamba	**31.63**	**22.03**	**57.91**	**45.19**

Table 8. Ablation studies on the multi-scale network in the main branch.

L_s	R@1		R@5	
	0.3	0.5	0.3	0.5
0	28.30	18.85	53.54	40.29
1	29.90	20.21	55.58	42.18
2	30.60	20.47	57.12	43.41
4	31.24	21.44	57.49	43.96
6	**31.63**	**22.03**	57.91	**45.19**
8	31.20	21.59	**58.06**	44.75

this experiment, R@1, 0.5 reaches 21.42%, and R@1, 0.3 reaches 31.22%, with an average drop of only 0.5 points. These results confirm that the performance improvement is not driven by any bias from movement-related verbs.

B.3. On Main Branch

Table 7 shows the impact of our self-mixer on the Ego4D-NLQ validation set. We opted Mamba over self-attention to enhance long-term temporal modeling, which results in a 2.37% improvement in R@1, 0.5. Additionally, Table 8 shows that a six-layer multi-scale network provides the best performance.

C. On Question Template

We conducted an in-depth analysis of our model's performance across various question templates in Ego4D-NLQ v2, as summarized in Table 9. The question templates are divided into three categories: queries about interacted objects, queries about background objects, and queries focusing on interactions involving people without specific objects. Queries with missing template information are classified as "None" and are relatively few.

From the table, we could find that the model's performance on questions involving background objects is significantly lower compared to the other two categories, highlighting the difficulty of understanding background elements in Ego4D-NLQ v2. Moreover, when comparing our model with GroundVQA [1], we observed an improvement of 1.39-8.08% in R@1, 0.5 for the background object categories, emphasizing that our enhancements significantly improve background object understanding.

D. Qualitative Experiments

Shot Visualization. We demonstrated the effect of shot segmentation in longer videos. As shown in Figure 1, camera movement is prevalent in egocentric videos, with the average shot length, using our shot-slicing strategy, being 13 seconds.

Ego4D-Goal-Step. As shown in Figure 2, our model accurately locates events in Ego4D-Goal-Step, demonstrating its strong ability to localize actions and objects.

Ego4D-NLQ. As shown in Figures 3, 4, and 5, our model accurately locates moments corresponding to various types of questions, demonstrating its robust and versatile localization capabilities.

E. Implementation Details

E.1. Model Structure

The text encoder consists of 4 transformer layers, the same number as the object encoder. The multi-modal fusion module contains $L_f = 4$ layers, while the multi-scale network has $L_s = 6$ layers. Additionally, the aggregators in the shot branch each use a single-layer network.

E.2. Object Detection

Popular object detectors like SAM and Grounding DINO were tested but struggled with detecting fine-grained objects. Therefore, Co-DETR was chosen, an open-source model that performs exceptionally well on LVIS, a dataset with over 1,000 object categories. To match objects in the query, we use spaCy to extract nouns and measure their semantic similarity to object classes. Of the 22,396 queries in the Ego4D-NLQ v2 dataset we used, 22,313 were found to contain nouns by spaCy, and 18,871 matched the object categories in LVIS.

E.3. Symbol t_{shot} of Figure 2

The blue line at the bottom represents the entire video timeline. The numbers 2 and 4, positioned below the red circles, indicate the segmentation timestamps corresponding to the 2nd and 4th narrations, which include movement-related verbs. Three lines above the blue line represent the three segmented shots.

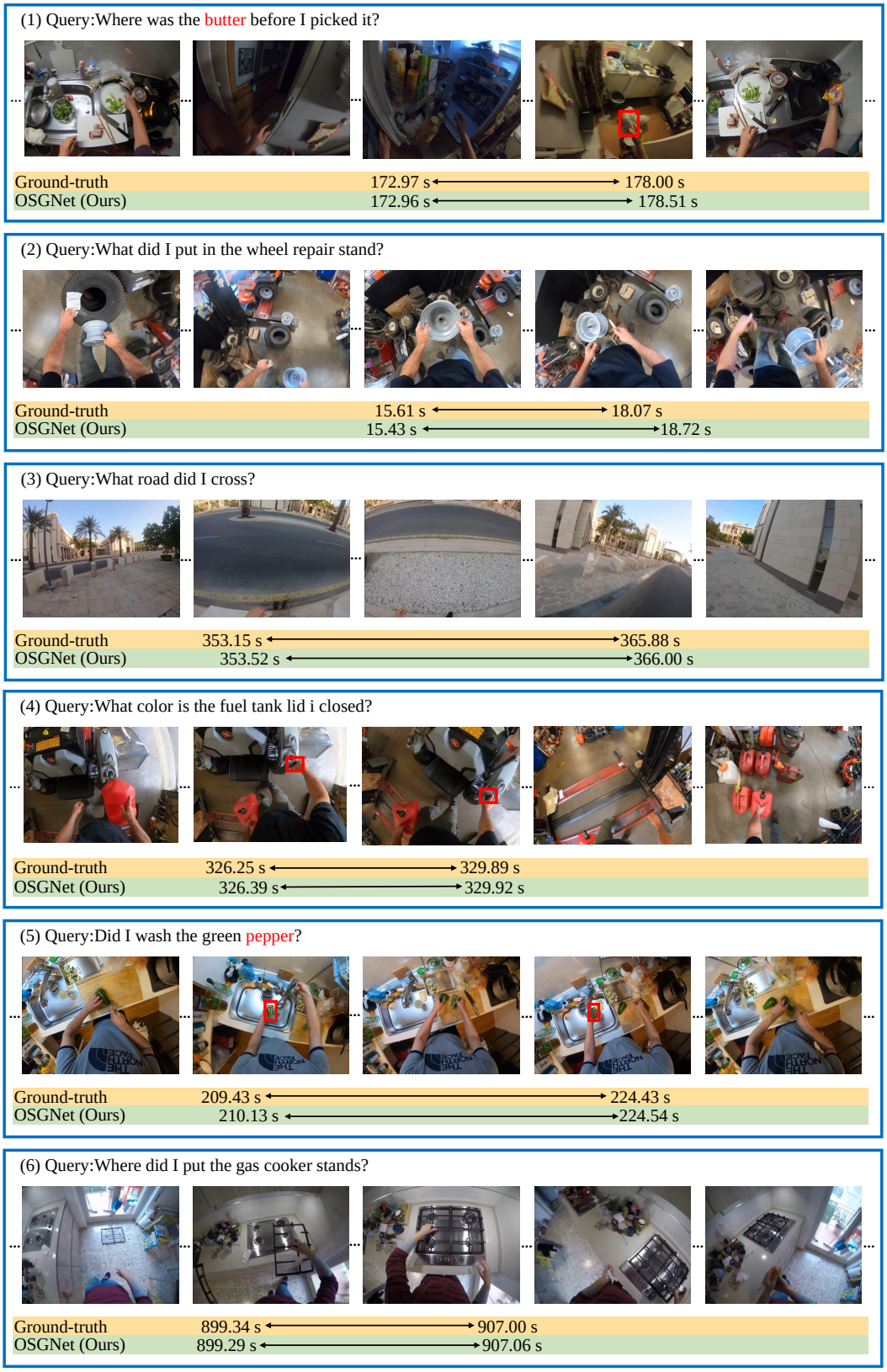

Figure 3. Illustration of grounding results on the question templates 1-6 in Ego4D-NLQ.

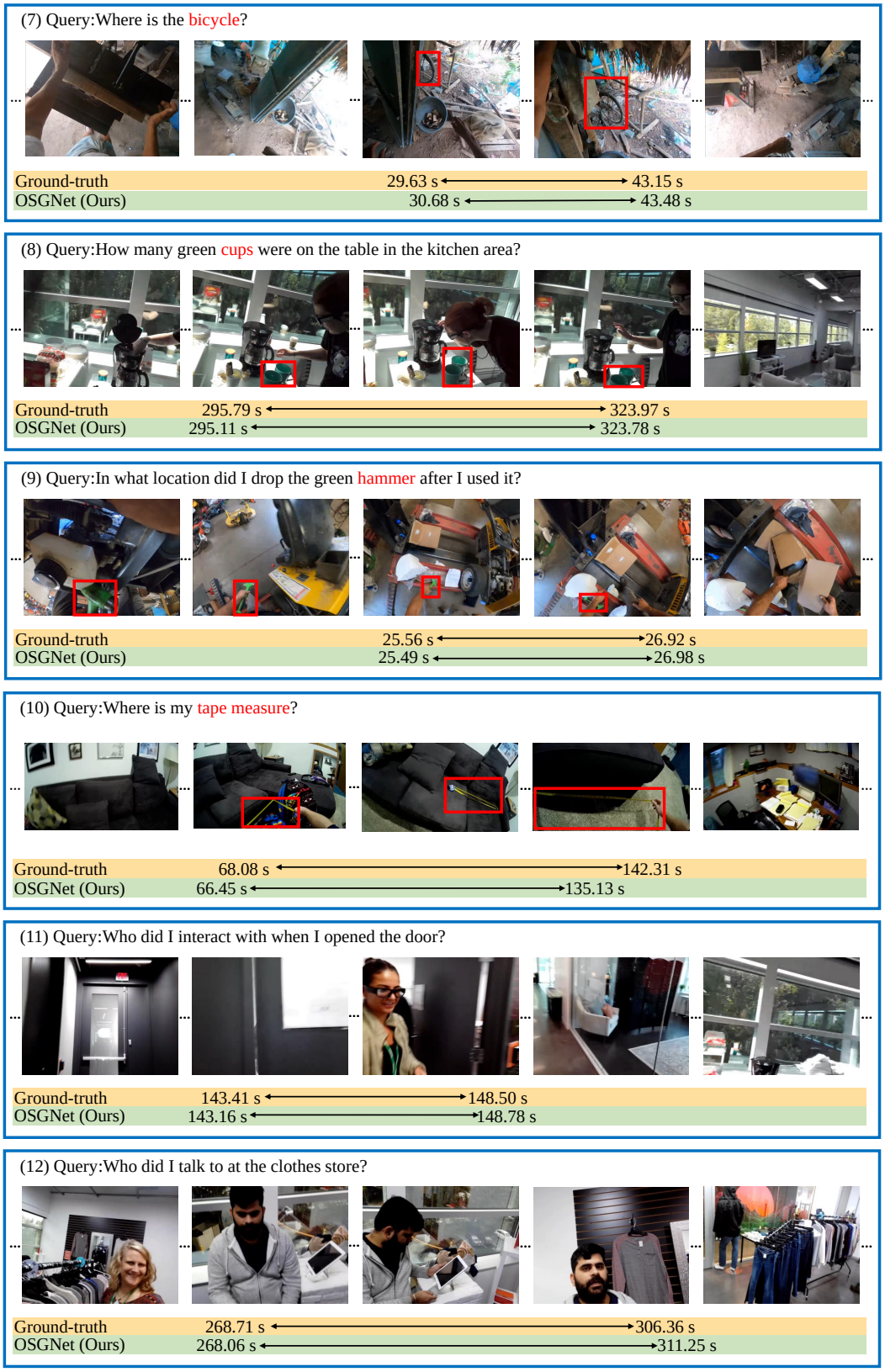

Figure 4. Illustration of grounding results on the question templates 7-12 in Ego4D-NLQ.

Table 9. Performance comparison on different question templates in Ego4D-NLQ v2.

Category	Template	No.	OSGNet(Ours)				GroundVQA			
			R@1		R@5		R@1		R@5	
			0.3	0.5	0.3	0.5	0.3	0.5	0.3	0.5
Interactive Objects	1. Where is object X before / after event Y?	750	37.07	23.33	64.53	48.93	40.67	25.33	63.73	44.93
	2. What did I put in X?	546	33.52	24.73	65.75	54.03	34.62	25.64	64.10	49.63
	3. What X did I Y?	350	39.14	28.86	66.86	54.00	38.57	28.29	67.43	50.29
	4. What X is Y?	332	26.81	17.77	48.19	37.65	26.20	17.17	42.47	28.31
	5. State of an object	235	42.55	30.21	70.21	56.17	39.15	26.38	60.43	43.83
	6. Where did I put X?	725	32.83	19.86	58.48	43.31	31.31	18.48	54.90	36.69
Background Objects	7. Where is object X?	552	18.30	14.49	42.75	30.98	11.59	7.97	29.35	18.66
	8. How many X's?	386	40.41	31.35	64.25	56.48	33.16	27.20	52.33	44.56
	9. In what location did I see object X?	359	20.33	15.60	43.45	33.70	11.70	7.52	31.75	23.12
	10. Where is my object X?	72	16.67	15.28	40.28	33.33	18.06	13.89	33.33	23.61
People	11. Who did I interact with when I did activity X?	115	31.30	20.00	53.91	39.13	26.96	15.65	53.04	33.91
	12. Who did I talk to in location X?	91	28.57	20.88	54.95	46.15	30.77	23.08	57.14	41.76
	13. When did I talk to or interact with person with role X?	22	22.73	13.64	54.55	36.36	18.18	13.64	50.00	22.73
None	14. None	17	29.41	23.53	58.82	35.29	35.29	35.29	47.06	47.06
Total		4552	31.63	22.03	57.91	45.19	29.68	20.23	52.17	37.83

E.4. Computational Efficiency

Pretraining takes 4 L20 GPUs for 3 days, while fine-tuning on Ego4D-NLQ requires 2 L20 GPUs for 3 hours. The model has 122M trainable parameters, but for inference, this reduces to 106M due to the shot branch being used only during training. On the NLQ v2 validation set, with an average video length of 9 minutes, inference speed is 19.45 video-text pairs per second using 1 L20 GPU. Text feature extraction with CLIP is very fast, processing thousands of sentences per second. Feature extraction times for LAVILA, InternVideo, and Co-DETR are 1/7, 1/3, and 1.5 times the video duration, respectively, using 1 L20 GPU.

References

[1] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12934–12943, 2024. 1, 3

[2] Tanveer Hannan, Md Mohaiminul Islam, Thomas Seidl, and Gedas Bertasius. RGNet: A unified retrieval and grounding network for long videos. In *European Conference on Computer Vision*, pages 1–23, 2024. 1

[3] Fangzhou Mu, Sicheng Mo, and Yin Li. SnAG: Scalable and accurate video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18930–18940, 2024. 1

[4] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. NaQ: Leveraging narrations as queries to supervise episodic memory. In *Proceedings of the IEEE/CVF Con-*

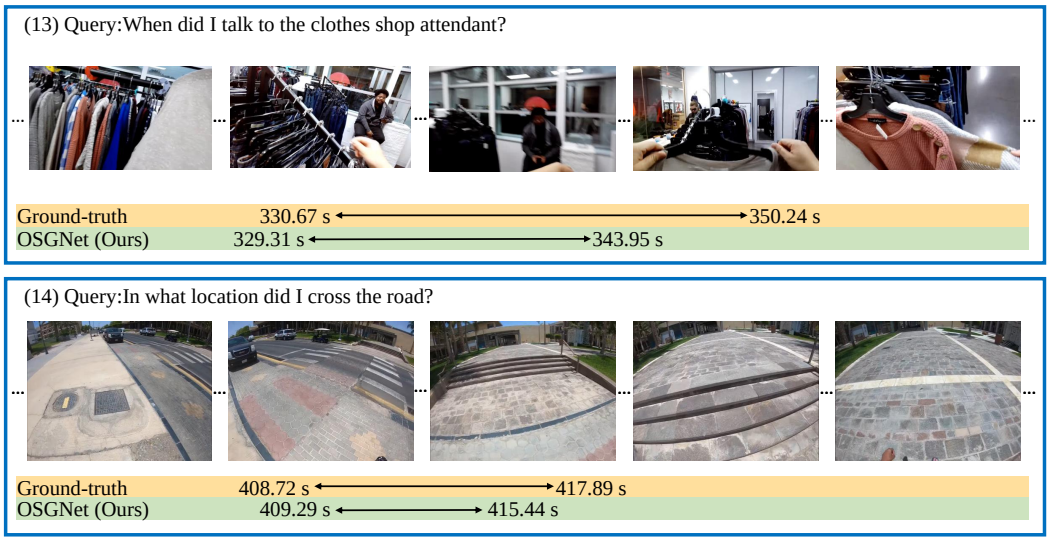

Figure 5. Illustration of grounding results on the question templates 13-14 in Ego4D-NLQ.

ference on Computer Vision and Pattern Recognition, pages 6694–6703. IEEE Computer Society, 2023. 2

[5] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023. 2