

Appendix

The appendix is organized as follows.

- **Dataset, Metric, Implementation Details:** We describe the datasets used, including highlight detection results on TVSum, the evaluation metrics employed, and key implementation details.
- **Token Dependency Analysis:** We examine the model’s reliance on the [EOS] token through attention correlation analysis, performing ablations under various token-conditioning settings with CLIP- and InternVideo2-based encoders.
- **Analysis of Phrase Segment:** We investigate how the optimal number of segmented phrases varies depending on the dataset and backbone features, analyzing its impact on model performance.
- **Ablation on Fusion Method:** We compare multiple strategies for fusing clip-level embeddings from the phrase and sentence paths.
- **Additional Visualization:** We include supplementary visualizations to better illustrate the model behavior and support claims made in the main paper.

A Dataset, Metric, Implementation Details

A.1 Dataset Description

QVHighlights QVHighlights ? is a large-scale benchmark for joint video moment retrieval and highlight detection. It contains 10,148 videos collected from YouTube, spanning various domains including daily life, travel, and news. Each video is paired with natural language queries and annotated with corresponding highlight segments.

Charades-STA Charades-STA ? extends the Charades dataset by adding temporal moment annotations aligned with text queries. It consists of 9,848 short videos depicting indoor human activities and provides 16,128 annotated query-moment pairs. The dataset is commonly used for evaluating moment retrieval performance and is provided with a standard train/test split.

TVSum TVSum ? is a video summarization dataset comprising 50 videos from 10 different categories such as documentary, sports, and travel. Each video is annotated with frame-level importance scores gathered through crowd-sourced annotations. Following prior work, we adopt a 4:1 train-test split and use video titles as textual queries in the highlight detection setting. Although originally intended for summarization, TVSum is widely repurposed for highlight detection due to the similarity between the two tasks.

A.2 Evaluation Metrics

We employ standard metrics commonly used in moment retrieval and highlight detection tasks. **Recall@1** is measured at multiple Intersection over Union (IoU) thresholds (e.g., 0.5 and 0.7), indicating whether the top-ranked prediction sufficiently overlaps with any ground-truth segment. **Mean Average Precision (mAP)** is computed by averaging the precision across multiple IoU thresholds, capturing both retrieval quality and temporal localization accuracy. **Hit@1** evaluates whether the top-scoring prediction exactly matches one of the ground-truth highlights, serving as a strict top-1 correctness measure. Additionally, **mean IoU (mIoU)** reports the average overlap between predicted and annotated segments.

We report Recall@1 (0.5/0.7), mAP, and Hit@1 on **QVHighlights**, Recall@1 (0.5/0.7) and mean IoU on **Charades-STA**, and top-5 mAP and Hit@1 on **TVSum**.

A.3 Experiment Results on TVSum Dataset

Table A1 presents the highlight detection performance on the TVSum *val* split across 10 video categories. Our method achieves the highest average mAP of **88.1**, outperforming existing baselines including TR-DETR and FlashVTG. Notably, our model exhibits strong consistency across Parade(PR), Attempting a Bike Trick(BT), and Dog Show(DS).

Table A1: Experimental results on the TVSum *val* dataset.

Method	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	Avg
LIM-S ?	55.9	42.9	61.2	54.0	60.4	47.5	43.2	66.3	69.1	62.6	56.3
Trailer ?	61.3	54.6	65.7	60.8	59.1	70.1	58.2	64.7	65.6	68.1	62.8
SL-Module ?	86.5	68.7	74.9	86.2	79.0	63.2	58.9	72.6	78.9	64.0	73.3
UMT ?	87.5	81.5	88.2	78.8	81.4	87.0	76.0	86.9	84.4	79.6	83.1
QD-DETR ?	88.2	87.4	85.6	85.0	85.8	86.9	76.4	91.3	89.2	73.7	85.0
UVCom ?	87.6	91.6	91.4	86.7	86.9	86.9	76.9	92.3	87.4	75.6	86.3
CG-DETR ?	86.9	88.8	94.8	87.7	86.7	89.6	74.8	93.3	89.2	75.9	86.8
TR-DETR ?	89.3	93.0	94.3	85.1	88.0	88.6	80.4	91.3	89.5	81.6	88.1
FlashVTG ?	88.3	94.3	91.5	87.7	87.1	91.1	74.7	93.4	90.3	81.7	88.0
DualGround	89.7	93.2	90.7	87.4	88.3	91.3	75.6	92.4	90.6	81.9	88.1

Table A2: Implementation details across datasets. From top to bottom, we list the hyperparameters and architectural configurations for QVHighlights (QVH.), Charades (Ch.), and TVSum (TVS.). In the **Feat** column, SF+C denotes the use of SlowFast and CLIP features, IV2 refers to InternVideo2, and I3D indicates I3D features. From left to right, **bs** is the batch size, **E** is the number of training epochs, and **lr** is the learning rate. **Ld** and **N** represent the counts of dummy tokens and phrase segments, respectively. **D.Enc** specifies the depth of dummy encoders, **ACA** is the number of adaptive cross-attention layers, and **P-SA** indicates the number of slot attention layers in the phrase-level path. **P.Enc** and **S.Enc** denote self-attention layers applied along the clip axis in the phrase-level and sentence-level paths, respectively. λ_{MR} , λ_{HD} , λ_{phrase} are loss weights for moment retrieval, highlight detection, and phrase-level supervision. r_{DQA} is a coefficient controlling the orthogonality regularization in the DQA loss.

Dataset	Feat	Hyperparameter					Layer #					Loss			
		bs	E	lr	Ld	N	D.Enc	ACA	P-SA	P.Enc	S.Enc	λ_{MR}	λ_{HD}	λ_{phrase}	r_{DQA}
QVH.	SF+C	64	150	$1e^{-4}$	3	4	2	3	2	2	2	5	1	1	0.3
QVH.	IV2	64	150	$1e^{-4}$	3	4	2	3	2	2	2	5	1	1	0.3
Ch.	SF+C	128	50	$2.5e^{-4}$	3	3	2	3	2	2	2	5	1	1	0.3
Ch.	IV2	128	50	$2.5e^{-4}$	3	3	2	3	2	2	2	5	1	1	0.3
TVS.	I3D	4	600	$1e^{-3}$	3	3	2	3	2	2	2	5	1	1	0.3

A.4 Implementation Details

Table A2 summarizes the training configurations across datasets. We vary the backbone features (SF+C, IV2, I3D) depending on the dataset and adopt consistent architectural settings. Specific hyperparameters, layer numbers, and loss coefficients are detailed in the table.

Each model uses a hidden dimension of 256 and is optimized with the AdamW optimizer. Transformer layers follow a post-norm architecture with 8 attention heads. For post-processing, non-maximum suppression (NMS) is applied with a threshold of 0.7. All experiments are conducted on a machine equipped with a Ryzen 3960X 24-core CPU and a single NVIDIA RTX 3090 GPU.

For the **InternVideo2 (IV2) ?** setting, we employ the pretrained model released by OpenGVLab. The video encoder corresponds to the 1B-parameter version of InternVideo2-stage2, while the text encoder is stage2-CLIP version (**InternVL-7B**) to enhance cross-modal representation quality. This configuration follows the official IV2-CLIP training pipeline and maintains consistent alignment between visual and textual embeddings.

B Token Dependency Analysis

We quantitatively evaluate the model’s dependency on the [EOS] token by measuring correlations of cross-modal attention pattern across tokens, which reveal the degree of over-reliance by the [EOS] token. We then analyze how varying textual token conditions [Word only, [EOS] only, and Full (Word + [EOS])] affect the performance of VTG models under two backbone settings: CLIP-based

Table A3: Performance of VTG models using the **SF+C** backbone across token conditions.

Method	Word	EOS	Full	R1@0.5	R1@0.7	mAP	mAP@0.5	mAP@0.75
CG-DETR	✓			64.84	49.68	43.18	65.27	44.16
CG-DETR		✓		62.19	46.13	41.87	64.01	42.43
CG-DETR			✓	66.90	50.32	43.47	65.48	44.79
TR-DETR	✓			66.32	50.45	43.99	65.74	44.89
TR-DETR		✓		64.00	47.74	41.78	64.25	42.45
TR-DETR			✓	66.48	50.71	44.53	65.43	44.98
FlashVTG	✓			68.85	53.81	48.42	67.83	51.50
FlashVTG		✓		65.62	52.60	45.32	67.12	50.49
FlashVTG			✓	<u>69.03</u>	54.06	<u>49.85</u>	<u>68.44</u>	<u>52.12</u>
DualGround	✓			68.20	<u>54.11</u>	48.51	68.02	51.83
DualGround		✓		65.91	52.31	45.44	67.24	50.33
DualGround			✓	69.25	54.87	49.96	68.62	52.30

Table A4: Performance of VTG models using the **IV2** backbone across token conditions.

Method	Word	EOS	Full	R1@0.5	R1@0.7	mAP	mAP@0.5	mAP@0.75
CG-DETR	✓			70.06	55.55	48.84	69.71	49.66
CG-DETR		✓		71.35	56.65	49.36	70.08	50.67
CG-DETR			✓	69.74	56.45	48.97	69.18	50.46
TR-DETR	✓			70.65	55.94	48.80	69.52	49.57
TR-DETR		✓		<u>73.35</u>	<u>58.84</u>	50.19	72.02	52.20
TR-DETR			✓	72.06	57.03	49.23	70.45	50.83
FlashVTG	✓			70.72	55.90	51.33	70.92	52.80
FlashVTG		✓		72.23	56.51	52.19	72.34	<u>55.60</u>
FlashVTG			✓	72.32	56.89	<u>52.26</u>	<u>72.39</u>	55.21
DualGround	✓			72.20	57.71	51.70	72.28	55.29
DualGround		✓		72.11	56.55	52.24	72.31	55.33
DualGround			✓	73.48	58.97	53.26	72.99	56.35

(SF+C) and InternVideo2-based (IV2). Table A3 and Table A4 present the moment retrieval results on the QVHighlights *val* set across these conditions.

B.1 Generalization of the EOS Over-Reliance

To verify that the over-reliance on the [EOS] token is prevalent phenomenon, we conduct a quantitative correlation analysis across the entire dataset. Specifically, we measure the statistical correlation between the attention weights assigned to the [EOS] token and those assigned to individual word tokens during cross-modal interaction. We adopt both the **Pearson** and **Spearman** correlation coefficients, which evaluate linear and rank-based relationships, respectively, to ensure robustness.

Pearson Correlation. Pearson correlation coefficient between two variables x and y is defined as:

$$r = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}, \quad (1)$$

where x_i and y_i denote individual data points, and \bar{x} , \bar{y} are their mean values. A higher r indicates a stronger linear relationship between x and y .

Spearman Correlation. Spearman rank correlation assesses monotonic relationships based on ranked values:

$$\rho = 1 - \frac{6 \sum_{i=1}^N d_i^2}{N(N^2 - 1)}, \quad (2)$$

where d_i is the rank difference between the paired values, and N is the number of data points.

Measurement Procedure. The overall computation process is summarized as follows:

1. **Cross-modal attention extraction:** For each sample in the training and validation sets, we extract the cross-modal attention map $\mathbf{A} \in \mathbb{R}^{N_t \times N_v}$, where N_t and N_v denote the number of text tokens and video clips, respectively.
2. **Token isolation:** We separate the attention vector of the [EOS] token, \mathbf{a}_{EOS} , and those of the remaining $N_t - 1$ word tokens $\{\mathbf{a}_i\}_{i=1}^{N_t-1}$.
3. **Token-wise correlation:** For each word token, we compute Pearson and Spearman correlations between \mathbf{a}_i and \mathbf{a}_{EOS} , yielding $(N_t - 1)$ correlation values per sample.
4. **Averaging:** We average the correlations across tokens and then across all samples within the subset, reporting mean Pearson and Spearman values for both training and validation sets.

Results. Table A5 summarizes the results for representative VTG models.

Table A5: Average Pearson and Spearman correlations between [EOS] and word-token attentions.

Model	Train		Val	
	Pearson	Spearman	Pearson	Spearman
CG-DETR	0.8960	0.8914	0.5962	0.7622
TR-DETR	0.8110	0.7753	0.6021	0.6340
FlashVTG	0.9745	0.9801	0.6771	0.7800

Discussion. Across all models, the correlation values remain consistently high (close to 1.0) for both Pearson and Spearman metrics, indicating that word tokens exhibit attention patterns highly similar to that of the [EOS] token. This confirms that prior VTG models show a generalized over-reliance on [EOS], where word-level semantics are largely overridden by global sentence-level alignment cues.

Furthermore, when relating these findings to the results in TableA4, we observe that models with weaker attention correlations tend to yield higher performance under the the single [EOS] token setting, when compared with Full-token setting. This suggests that in current model architectures, suppressing the local semantic contributions of individual word tokens may lead to a more optimized training trajectory.

B.2 Impact of Backbone Semantics

Under the SF+C backbone (Table A3), using the [EOS] token alone consistently yields lower performance than using word tokens across all models. In contrast, the IV2 backbone (Table A4) shows the opposite trend: in all models except ours, the [EOS]-only setting achieves either the best performance (e.g., CG-DETR, TR-DETR) or results comparable to other configurations (e.g., FlashVTG).

We attribute this discrepancy to the difference in feature dimensionality between the backbones. CLIP encodes each token as a 512-dimensional vector, while InternVideo2 produces 4096-dimensional embeddings. This higher capacity allows IV2’s [EOS] token to carry richer sentence-level semantics, enabling strong alignment even without word-level information. Conversely, CLIP’s limited [EOS] capacity cannot fully represent complex queries, leading models to fall back on word tokens for localized cues. However, this reliance arises not from an intentional design but as a side effect of the [EOS] token’s limitations. Treating all tokens uniformly in a flat sequence still ignores their distinct semantic roles, leading to suboptimal alignment.

Our proposed method alleviates this issue by separating sentence-level and phrase-level semantics. As shown in Table A3, it achieves robust performance even with CLIP-based features, validating its effectiveness despite the limited capacity of the [EOS] token. As vision-language models (VLMs) evolve with increasingly powerful text encoders, the [EOS] token will likely play an even greater role, making proper treatment of token-level semantics a critical consideration for future VTG architectures.

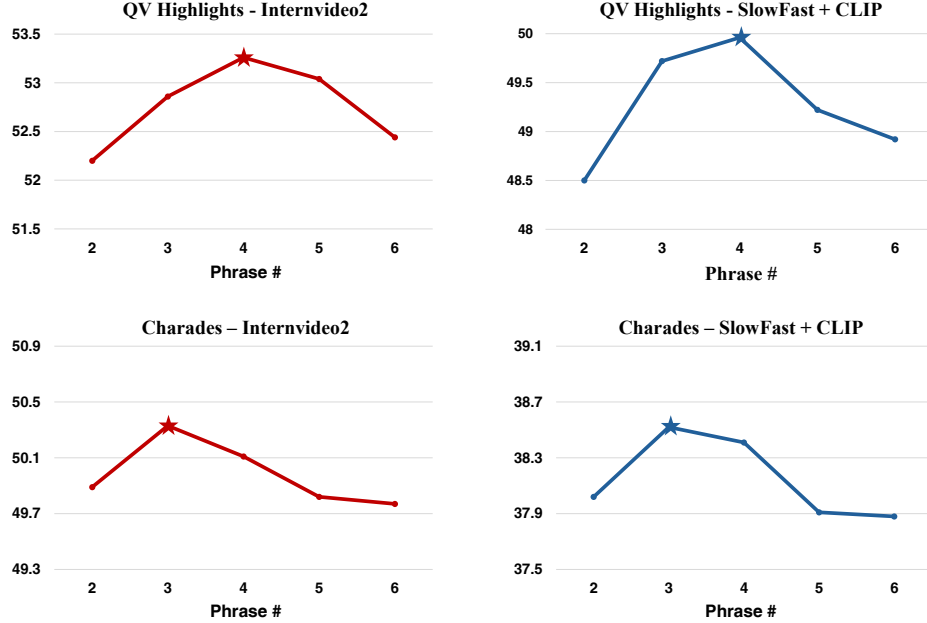


Figure A1: Ablation study on the number of phrase segments.

Table A6: Average query length per dataset.

Dataset	Split	Query Length
QVHighlights	train	10.46
	val	10.49
Charades	train	6.21
	val	6.23
TVSum	train	7.55
	val	7.70

Table A7: Ablation on Fusion Method

Option	R1@0.5	R1@0.7	mAP
Add	<u>73.48</u>	58.97	53.26
Hadamard	71.71	55.25	51.27
Gate	73.51	<u>58.71</u>	<u>53.24</u>
Concat-mlp	73.66	58.19	52.91

B.3 Architectural Influence on Token Utilization

As shown in Table A4, CG-DETR and TR-DETR achieve better performance when using only the [EOS] token, compared to word or full token inputs. This suggests that word tokens may act as noise in these architectures. In CG-DETR, the clip-word distillation loss emphasizes alignment with individual words, which can suppress the rich global semantics of the [EOS] token. In TR-DETR, the global textual feature used for regulation is computed by mean-pooling over all word tokens. This strategy may dilute the semantic strength of the [EOS] token and introduce noise from weakly aligned or irrelevant words. In both cases, using only the [EOS] token avoids such noise and leads to better alignment.

These results suggest that the integration of token-level inputs should account for the distinct semantic roles of word and [EOS] tokens. Word tokens are most effective when they complement the global sentence representation without interfering with it. Our DualGround framework supports this balance by explicitly disentangling global and local semantics.

C Analysis of Phrase Segment

C.1 Ablation on Phrase Segment Number

To determine the optimal number of phrase segments, we conduct an ablation study on the phrase segmentation parameter N , which defines the number of semantic units extracted from the input

query. We experiment with different values of N on both the Charades and QVHighlights datasets using two backbones: **SlowFast + CLIP** and **Internvideo2**.

As shown in Fig. A1, Charades achieves the best performance at $N = 3$, while QVHighlights yields the highest accuracy at $N = 4$. This difference is further analyzed in the next subsection. We also observe a performance drop when N becomes large. Excessive segmentation divides queries into overly short spans, which may fail to capture complete semantic units and lead to fragmented or diluted phrase representations. This prevents effective alignment with video content and undermines the benefits of phrase-level modeling.

C.2 Effect of Query Complexity

We hypothesize that the optimal number of phrase segments is influenced by the complexity of text queries. Intuitively, queries with greater semantic richness benefit from finer phrase decomposition, as they contain more diverse word-level information that can be aligned with visual content.

To investigate this, we analyze the average query length across datasets, as shown in Tab. A6. Queries from QVHighlights are substantially longer than those from Charades or TVSum, indicating higher semantic complexity. This aligns with our ablation results, where QVHighlights achieves the best performance at $N = 4$, while Charades performs best at $N = 3$. These observations suggest that phrase segmentation should be tailored to the dataset’s linguistic characteristics.

D Ablation on Fusion Strategy

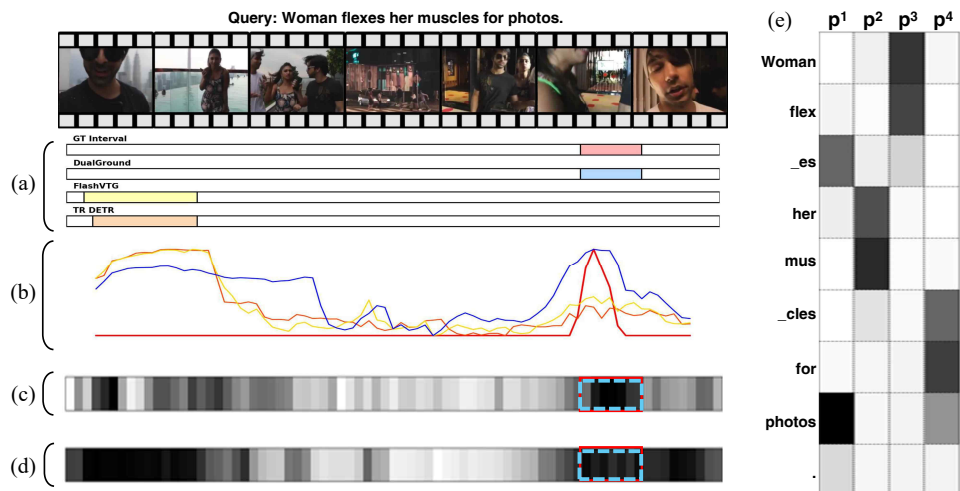
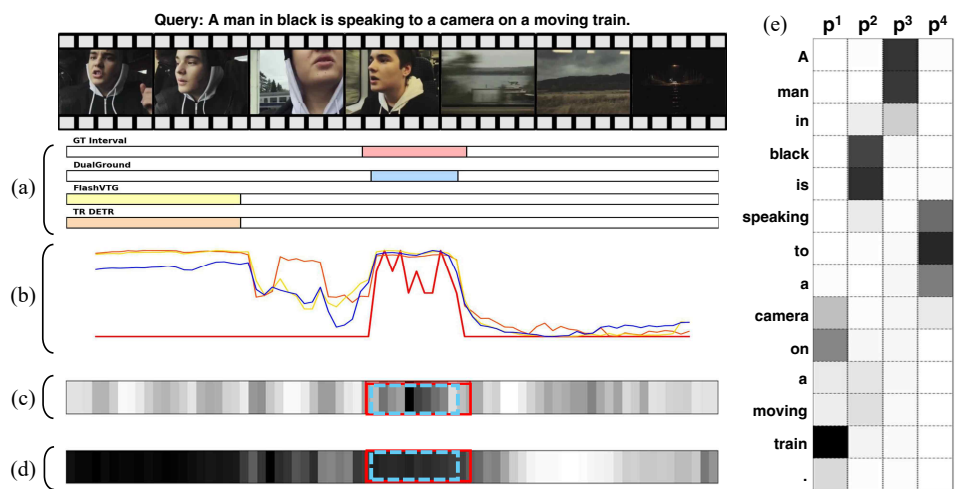
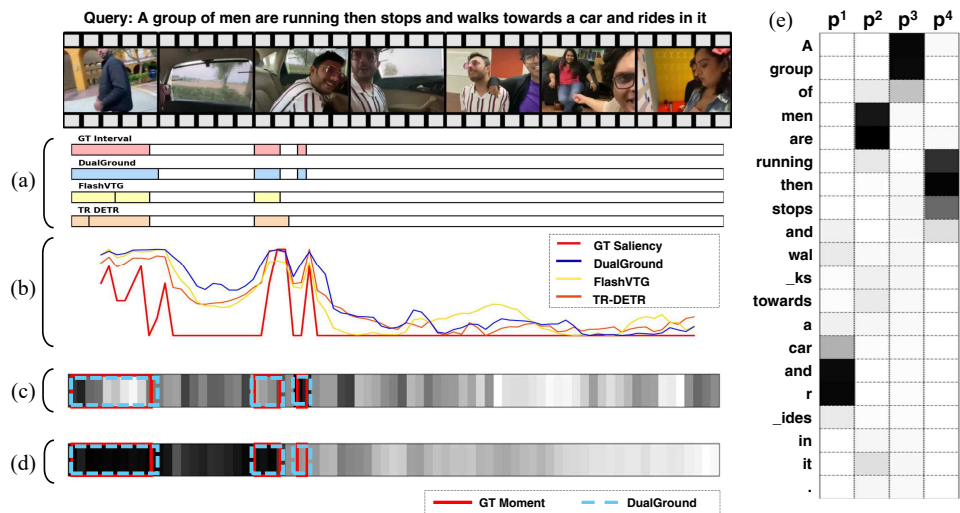
We evaluate four different strategies for integrating the sentence-level (V_s) and phrase-level (V_p) features into a unified representation $F = V_s + V_p$, which is used for downstream prediction (see Sec. 3.4). The following options are compared in Tab. A7:

- **Add**: Element-wise addition of V_s and V_p . This is our default configuration due to its simplicity and efficiency.
- **Hadamard**: Element-wise multiplication of V_s and V_p , emphasizing shared dimensions.
- **Gate**: A learnable sigmoid gate σ is applied such that the fused feature $F = \sigma \cdot V_s + (1 - \sigma) \cdot V_p$. This allows the model to adaptively weight sentence and phrase contributions.
- **Concat-mlp**: The two features are concatenated and passed through a linear projection layer to match the original dimensionality.

As shown in Tab. A7, the **Add** method achieves the best overall performance considering both effectiveness and computational simplicity. While *Concat-mlp* slightly improves R1@0.5, its performance on R1@0.7 and mAP is inferior to *Add*. The *Gate* mechanism performs comparably but introduces additional parameters and complexity. We thus adopt **addition** as our default fusion strategy due to its favorable trade-off between accuracy and efficiency.

E Additional Visualization

We provide additional qualitative results in Figure A2. The visualizations demonstrate how semantically aligned word tokens are clustered into meaningful phrases, as illustrated in A2(e). This grouping provides localized, clip-wise information that complements the global sentence-level representation, particularly in cases where fine-grained cues are difficult to capture. As a result, it enables more accurate and context-aware temporal grounding.



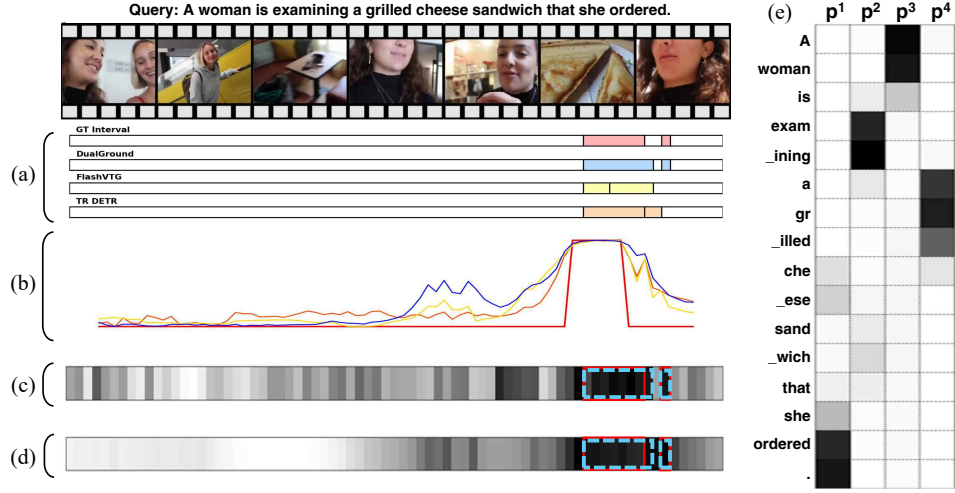


Figure A2: Additional Visualization results on the QVHighlights validation split. (a) Moment retrieval predictions and (b) Highlight detection scores are compared across models. (c) L2 norm activation map of phrase-level embeddings, (d) L2 norm activation map of sentence-level embeddings, and (e) Phrase-to-word attention map are visualizations from our proposed DualGround model, highlighting how it captures localized semantics and structured alignment.