Narrating the Video: Boosting Text-Video Retrieval via Comprehensive Utilization of Frame-Level Captions

Supplementary Material

The supplementary is organized as follows:

- The details of the multi-grained matching method used for query-video-narration alignment in Sec. A.
- Further information on the datasets used in the experiments, along with additional implementation details, in Sec. B.
- The details of the frame-level caption generation process in Sec. C.
- Additional ablation studies in Sec. D.
- Further qualitative analyses for each dataset in Sec. E.

A. Multi-Granularity Matching Module

Inspired by previous research [40], we propose a modified multi-granularity matching process with our proposed narration and the result of a query-aware adaptive filtering module.

Coarse-grained matching. In Fig. 5 (a), we illustrate the coarse-grained matching process. For both video and narration, the sequence of embedding vectors for frames and captions is transformed into a single vector through weighted pooling. The cosine similarity between this pooled vector and the $w_{[EOS]}$ vector of the query is then computed to obtain the coarse matching scores $s_{coarse}(\boldsymbol{q}, \boldsymbol{z}^v)$ and $s_{coarse}(\boldsymbol{q}, \boldsymbol{z}^n)$. Here, it should be remarked that the weight values for the pooling are determined by the query-aware filtering module. In the query-aware adaptive filtering module described in Sec. 3.4, we evaluate individual frames and captions based on their similarities to the given query, and obtain the final filtering scores for the selected frames $\boldsymbol{z}^v$ and captions $\boldsymbol{z}^n$ through filtering and softmax. An example of the filtering scores is shown in Fig. 2 (a).

By using the filtering scores as weights for pooling, we can emphasize the most relevant frames in the query-video matching process and the most relevant captions in the query-narration matching process.

Fine-grained matching. Fig. 5 (b) shows the details of the proposed fine-granularity matching process. For query-video matching, we first compute a similarity matrix with the cosine similarity of all possible pairs of frame embeddings and word embeddings and take frame-wise and word-wise maximums to get two vectors of maximum similarity values. The frame-wise maximum similarity vector is then used to compute the inner product with the filtering score to obtain the similarity score s_{W2F}. We also compute a similarity score s_{F2W} using a word-wise maximum similarity vector and a weight vector from an FC layer that

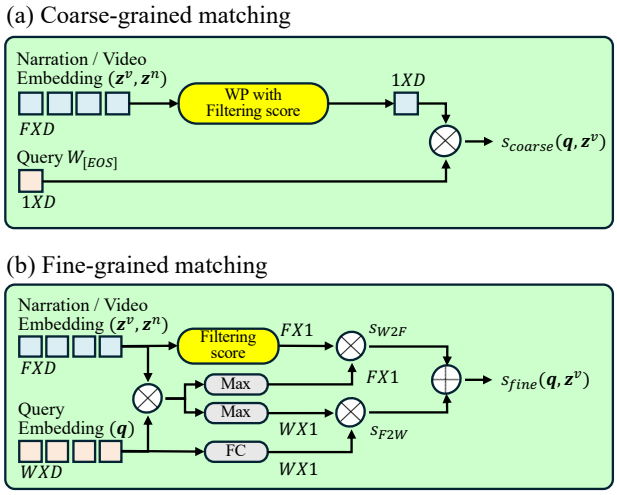

Figure 5. Process of coarse-grained and fine-grain matching. In the yellow box, we utilize the results of nucleus filtering for weight values.

is trained in an end-to-end manner. Finally, the two similarity scores are summed to obtain a fine-granularity score $s_{fine}(\boldsymbol{q}, \boldsymbol{z}^v)$. For query-narration matching, the same process is performed with narration embedding vectors to obtain the score $s_{fine}(\boldsymbol{q}, \boldsymbol{z}^n)$.

B. Experimental Settings

B.1. Datasets

To validate our model, we utilized the MSR-VTT [41], MSVD [3], VATEX [36], and DiDeMo [1] datasets, which are commonly employed in previous studies. Tab. 6 shows a summary of the properties of four datasets.

MSR-VTT provides 10K web video clips, totaling 38.7 hours of content, along with 200K clip-sentence pairs. These clips, collected from the YouTube platform, cover a wide range of visual content across various comprehensive categories. Each clip ranges from 10 to 30 seconds in length and is annotated with approximately 20 natural sentences generated by 1,327 Amazon Mechanical Turk (AMT) workers. MSR-VTT is typically divided into three types of splits (training/testing): full split (7K/3K), 1k-A split (9K/1K) [44], and 1k-B split (6.5K/1K) [28]. Among these, the 1k-A split is most commonly used for performance comparison, and we adopted this in our evaluation.

The MSVD dataset consists of over 85,000 English descriptions for 2K video snippets. These descriptions were

Dataset	Videos	Captions	Length	Source
MSR-VTT [41]	10K	200K	10-30s	Youtube
MSVD [3]	2K	8.5K	4-10s	Youtube
VATEX [36]	35K	700K	Avg 10s	Kinetics -600, Youtube
DiDeMo [1]	10K	-	Max 30s	Flickr

Table 6. Statistics of four benchmark datasets.

generated by AMT workers who summarized the action in each short video snippet with a single sentence. The video clips are generally set to a short length of 4 to 10 seconds, each aiming to depict a clear event or action.

VATEX comprises over 35K videos, each averaging 10 seconds in length, accompanied by 700K captions in both Chinese and English. This dataset includes more than 206,000 parallel English-Chinese translations, featuring long sentences with relatively diverse lexical characteristics.

DiDeMo consists of videos sourced from Flickr, which are trimmed to a maximum length of 30 seconds and divided into 5-second segments to reduce annotation complexity. The dataset is split into training, validation, and test sets containing 8,395, 1,065, and 1,004 videos, respectively. In total, the dataset includes 26,892 moments, with each moment potentially associated with multiple descriptions by different annotators. Following the approach of previous studies [25, 40], we concatenated all sentence descriptions of each video into a single sentence query for evaluation purposes.

B.2. Implementation Details

To ensure reproducibility, we provide additional implementation details in this section. Following the settings of CLIP4Clip [25], we configure the initial learning rate to 1e-7 for the CLIP [30] module, while other modules are set to 1e-4. The Adam optimizer is utilized, accompanied by a cosine scheduling strategy and a warm-up proportion of 0.1. Additionally, the temperature hyperparameter for all softmax functions is set to 0.1.

For preprocessing all datasets, we follow the data preparing process of CLIP4Clip. Also, we uniformly sample each video data at 12 frames for MSR-VTT, MSVD, and VATEX datasets, and sample 32 frames for DiDeMo to cover longer video duration.

For the hard negative loss, the hyperparameter settings vary by dataset. For MSR-VTT with the ViT-B/32 backbone, we set $\lambda = 0.7$, $\eta = 1.8$, and $\alpha = 1$. In contrast, for the ViT-B/16 backbone, which demonstrates better performance, we focused on hard negative selection and loss weighting for performance optimization. Accordingly, for MSR-VTT, we set $\lambda = 1.1$, $\eta = 2.0$, and $\alpha = 2$. For MSVD, the values are $\lambda = 0.9$, $\eta = 1.8$, and $\alpha = 1$. For VATEX, we use $\lambda = 0.9$, $\eta = 1.8$, and $\alpha = 0.8$. For DiDeMo, we set $\lambda = 1.0$, $\eta = 1.9$, and $\alpha = 1$. All experiments are conducted on two NVIDIA RTX A6000 GPUs to ensure consistency in our experimental setup.

Type	Prompt & Characteristic
A	Usage Prompt: *Briefly describe the object in the image in one sentence.* Characteristic: Simple, Single sentence
B	Usage Prompt: *You want to view an image, one of the frames in a video clip, and organize the text description into a single sentence. Follow these steps. 1. Analyze the input image by dividing it into objects, actions, and other parts. 2. Create a one-sentence text description based on your previous analysis. 3. Output only the processed text without any additional description.* Characteristic: Structured, Single sentence
C	Usage Prompt: *Please describe this image for image-captioning task* Characteristic: Simple, Not single sentence

Table 7. Three different types of prompts used for caption generation. The characteristic of each type is also denoted.

Prompt type	Text-to-video retrieval		
	R@1	R@5	R@10
A	52.6	**82.0**	**89.5**
B	52.6	81.6	89.0
C	**53.1**	81.4	88.8

Table 8. Text-to-video retrieval results on MSVD depend on the types of prompts used for caption generation.

C. Caption Generator

C.1. Prompt Strategies

When using large multimodal models (LMMs) for caption generation, both the caption quality and inference time can vary depending on the prompt. Although our NarVid

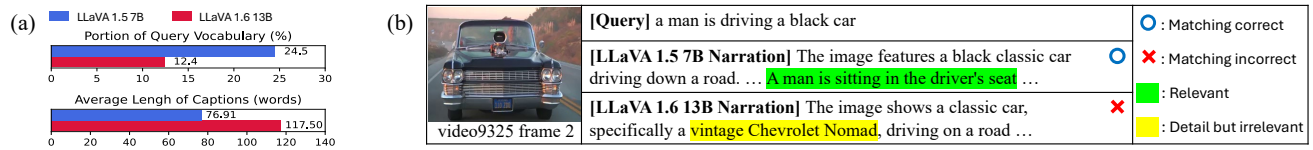

Figure 6. Narration comparison on MSR-VTT across different LLaVA versions.

Captioner	Time(s)	Text-to-video retrieval		
		R@1	R@5	R@10
LLaVA 0.5B	**7.92**	50.0	76.2	84.9
LLaVA 7B	25.31	**51.0**	**76.4**	**85.2**

Table 9. Comparison of text-to-video retrieval results on MSR-VTT based on the LLaVA model size.

framework generates narrations offline, the computational resources and time required for pre-generating captions remain practical challenges in our experiments. Finding the optimal prompt for each dataset requires substantial experimentation, which may be beyond the scope of our current research. To address this challenge, we utilized the MSVD dataset [3] as a testbed, which requires the least computational resources for inference among the datasets. We first evaluated the retrieval performance across various prompts on MSVD and then applied our findings to other datasets.

As shown in Tab. 7, we provide several examples of prompts with their specific characteristics. We used the LLaVa 1.5 7B model [22] as a baseline model for the experiments. Prompt A is a simple request to describe the objects within each frame in a single sentence. Prompt B is more structured, requiring a single-sentence description that includes objects, actions, and other parts. Prompt C removes the single-sentence constraint, allowing for a more detailed description of each frame. As shown in Tab. 8, prompt C achieves the highest performance in R@1, and we use it for other experiments. Given the apparent potential for improvement in these captions, we expect that various advanced inference techniques, such as in-context learning, will enhance the performance in future work. The captions used for experiments will be publicly available with our code.

C.2. Time Efficiency for Caption Generation

In our experiments, we generated all frame-level captions per video offline. As shown in Tab. 9, we mainly use the basic LLaVa 1.5 7B model as a captions generator, therefore the inference time can be reduced with inference boosting toolkits. One interesting point is that a lighter frame captioner (0.5B) can achieve considerable performance with less computing time.

Matrix	Text-to-video retrieval (zero-shot)				
	R@1	R@5	R@10	MdR	MnR
S_{qv}	**31.4**	**53.8**	**62.5**	**4.0**	**41.3**
S_{qn}	13.7	25.6	31.5	64.5	188.9
S_{sum}	21.5	37.3	43.8	15.0	109.9
S_{fusion}	27.3	47.0	56.1	6.0	63.9

Table 10. Results depending on similarity matrices on MSR-VTT for zero-shot retrieval evaluation. Setting: we extracted CLIP features from the videos and their narrations to compare with the queries' CLIP features without any training. S_{qv} and S_{qn} are the query-video and query-narration similarity matrices, each. S_{sum} indicates the element-wise sum of the two matrices, while S_{fusion} represents the element-wise sum of the two normalized matrices.

Matrix	Text-to-video retrieval					
	MSR-VTT			DiDeMo		
	R@1	R@5	R@10	R@1	R@5	R@10
S_{qv}	46.3	74.3	82.9	48.9	77.7	84.8
S_{qn}	46.0	75.2	83.2	46.7	72.9	82.2
S_{sum}	51.0	76.4	85.2	52.5	79.1	85.7
S_{fusion}	**51.0**	**76.4**	**85.2**	**53.0**	**79.5**	**86.0**

Table 11. Retrieval performance comparison based on the similarity matrices on our NarVid framework. Note that S_{qv}, S_{qn} S_{sum} and S_{fusion} are same operation as Tab. 10.

C.3. Performance Discrepancy by LLaVA 1.5 7B vs 1.6 13B

In general, the larger VLM tends to generate more detail and long captions, but all of them are not necessarily related to given queries, which shows less portion of query vocabulary (Fig. 6 (a)). The well-generated captions with query-irrelevant words may cause a negative effect on the retrieval performance, as shown in Fig. 6 (b).

D. Ablation Study

D.1. Effectiveness of Narration Utilization

Zero-shot retrieval performance. *Can captions from good models such as LMMs deliver outstanding results on their own?* To answer this question, in Tab. 10, we evaluate zero-

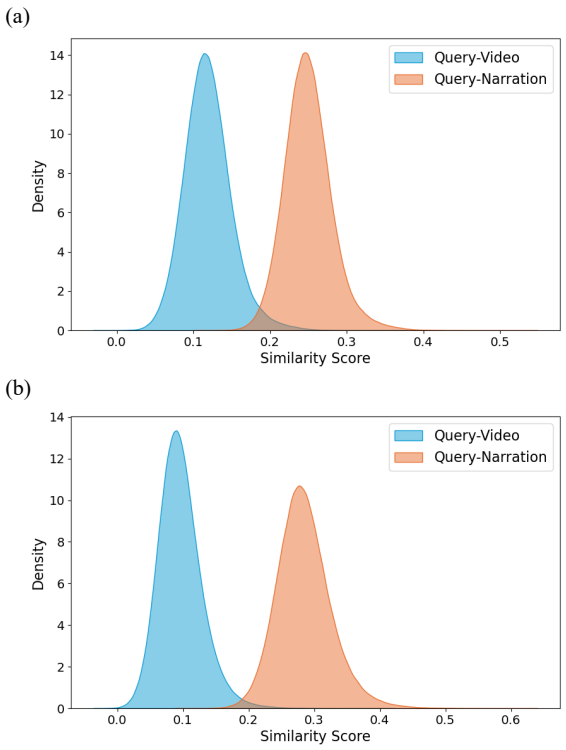

Figure 7. Visualization of similarity matrix distribution for query-video and query-narration. (a) results for MSR-VTT. (b) results for DiDeMo

shot retrieval to verify the importance of effectively utilizing narration information without training. For the experiment setting, we extracted CLIP features from each video and its narration, applied mean pooling, and evaluated zero-shot retrieval performance with given queries. In the table, the query-to-video retrieval shows reasonable retrieval performance without training, while the query-to-narration retrieval shows significantly lower performance. Although an element-wise matrix summation or the fusion approach used in our NarVid framework shows some improvements, the retrieval performances remain insufficient for practical retrieval scenarios. These experimental results once again support the claim of our framework that the information from the generative model should be utilized appropriately, including training.

Similarity matching fusion. After the training process on our NarVid, we utilize the query-video similarity matrix $\boldsymbol{S}_{qv}$ as well as the additional query-narration similarity matrix $\boldsymbol{S}_{qn}$ during the inference phase. However, as shown in Fig. 7, the distributions of the elements in the two matrices are notably different. Therefore, a simple summation of these matrices could lead to an overemphasis on one aspect over the other. To mitigate this issue, we standardize the element values in each matrix separately before summation, using their means (μ^{qv}, μ^{qn}) and standard deviations $(\sigma^{qv}, \sigma^{qn})$. The two standardized matrices are then fused to form the final score matrix $\boldsymbol{S}_{fusion}$, ensuring a balanced consideration of both aspects, which is defined in Eq. (11).

Temporal modeling		Text-to-video retrieval		
$\phi_{temp}(\hat{\boldsymbol{v}})$	$\phi_{temp}(\hat{\boldsymbol{n}})$	R@1	R@5	R@10
	✓	49.1	75.9	84.8
✓		50.2	75.8	85.4
✓	✓	51.0	76.4	85.2

Table 12. Comparison of the impact of temporal modeling on performance across modalities on MSR-VTT

Tab. 11 presents the effect of the similarity matrix fusion on the MSR-VTT [41] and DiDeMo [1] dataset. The improvement of over 4.4% in R@1 performance when combining the two matrices indicates that the information from video and narration works complementarily. Furthermore, the results on DiDeMo confirm that $\boldsymbol{S}_{fusion}$ outperforms the element-wise summation. The results demonstrate the effectiveness and robustness of our fusion method, especially when the distributions of the two matrices differ regarding their standard deviations, as illustrated in Fig. 7 (b).

D.2. Temporal Modeling Details.

The temporal block in Eq. (3) is defined as $\phi_{temp}(\hat{\boldsymbol{v}}) = Transformer(\hat{\boldsymbol{v}} + \boldsymbol{P}) + \hat{\boldsymbol{v}}$, where $\boldsymbol{P}$ represents the positional embedding, as used in CLIP4Clip [25]. Temporal modeling is commonly applied to videos, as supported by Row 1 of Tab. 12. Similarly, although narrations have inherent temporal orders, we believe that effectively capturing the temporal relationships among frame-level captions enhances representation. Notably, removing ϕ_{temp} from the narration led to a 0.8% decrease in R@1, highlighting the significance of temporal modeling.

D.3. Cross-View Hard Negative Loss

Fig. 8 shows the performance of R@1 with varying values of λ and η, which are hyperparameters used in the hard negative rank loss of Eqs. (6) and (8). λ is a scaling factor that establishes the threshold for selecting hard negatives within the query-video and query-narration similarity matrices. Likewise, η serves as a scaling factor that modifies the thresholds in the cross-view hard negative loss. The balance between λ and η is crucial for optimal performance, achieving its best results at $\lambda = 0.7$ and $\eta = 1.8$.

Intuitively, employing the same hyperparameters during training may lead to a decrease in the selection of hard negative samples, consequently diminishing the impact of the hard negative loss. This raises an intriguing question about the benefits of focusing on hard negative samples in the later

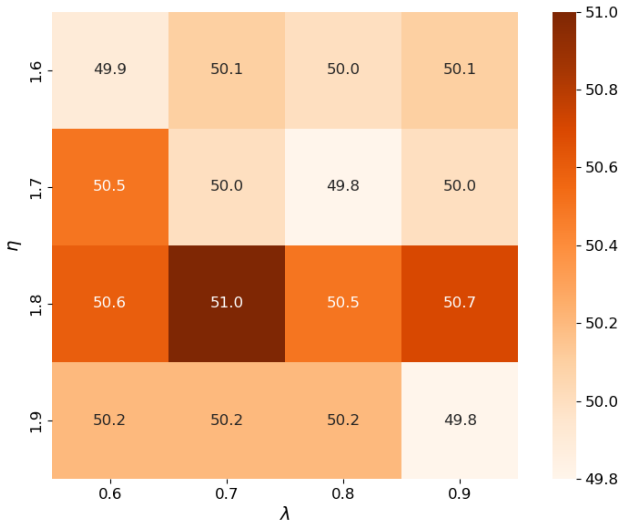

Figure 8. Ablation study results of λ and η, used in the hard negative loss, with α set to 1, conducted on the MSR-VTT.

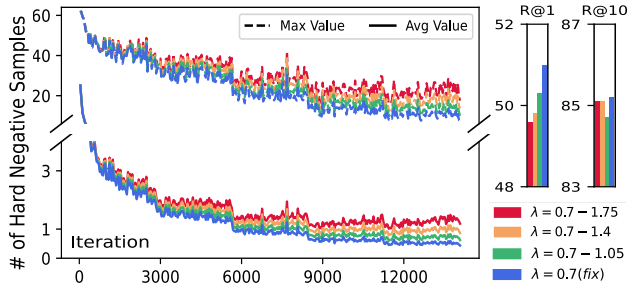

Figure 9. Variations in selected hard negatives with the increase of λ, along with the corresponding recall results on MSR-VTT.

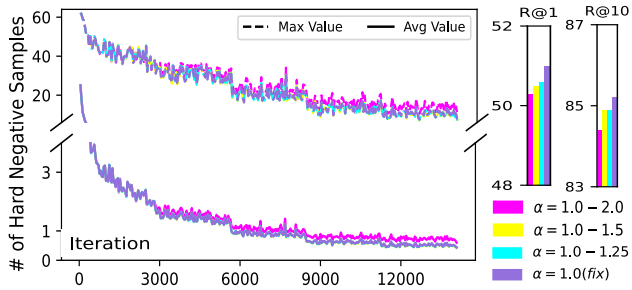

Figure 10. Variations in selected hard negatives with the increase of α, along with the corresponding recall results on MSR-VTT.

stages of training. To address this question, we linearly increased λ to acquire more hard negative samples during experimentation. As shown in Fig. 9, our findings reveal that while this approach successfully increased the quantity of hard negative samples, it resulted in a degradation of performance. This indicates that forcing easy samples to be treated as hard negatives may hinder representation learning.

Loss	Text-to-video retrieval		
	R@1	R@5	R@10
Only L_{NCE}	50.4	76.1	84.6
$L_{NCE} + L^{H}_{NCE}$	50.9	76.2	85.2
$L_{NCE} + L_{CVH}$	**51.0**	**76.4**	**85.2**

Table 13. Comparison of performance results for different loss configurations on MSR-VTT.

Fig. 10 presents the results of linearly increasing α, which similarly shows performance degradation. Notably, in the case of increasing α from 1.0 to 2.0, which emphasizes the hard negative loss, we found that the number of selected hard negative samples actually increased. The result suggests that dynamically modifying the hard negative loss weight to focus more on hard negatives may negatively impact the learned representations.

We conducted additional experiments with different types of hard negative loss functions; using the same unified hard negative samples H_i (for query-to-video and query-to-narration) and H^T_i (for video-to-query and narration-to-query), we evaluate the performance of hard negative loss based on the InfoNCE loss, L^H_{NCE}, defined as follows:

$$L^H_{qv} = -\frac{1}{2B}\sum_{i=1}^{B}\left\{\log\frac{e^{\boldsymbol{S}_{qv}(i,i)}}{\sum_{j\in H_i} e^{\boldsymbol{S}_{qv}(i,j)}} + \log\frac{e^{\boldsymbol{S}_{qv}(i,i)}}{\sum_{j\in H^T_i} e^{\boldsymbol{S}_{qv}(j,i)}}\right\},$$

$$L^H_{qn} = -\frac{1}{2B}\sum_{i=1}^{B}\left\{\log\frac{e^{\boldsymbol{S}_{qn}(i,i)}}{\sum_{j\in H_i} e^{\boldsymbol{S}_{qn}(i,j)}} + \log\frac{e^{\boldsymbol{S}_{qn}(i,i)}}{\sum_{j\in H^T_i} e^{\boldsymbol{S}_{qn}(j,i)}}\right\},$$

$$L^H_{NCE} = \frac{1}{2}(L^H_{qv} + L^H_{qn}). \tag{12}$$

As shown in Tab. 13, both types of hard negative loss assist the model in learning discriminative features. This highlights the effectiveness and robustness of our method that leverages unified negative samples from two different views: inter-modality (query-video) and intra-modality (query-narration).

D.4. Cost per Each Module.

For the number of trainable parameters, the two Cross-Modal Interactions have $\phi_{co-attn}$ (3.7M) and ϕ_{temp} (12.7M), respectively. The Narration Matching has 0.3M, leading to a 12% increase over the baseline [25] (136.2M).

For FLOPs, the Narration Matching shows a 30.5G increase due to the expanded caption encoding, reaching 1.5 times the baseline (54.4G). The others have little impact.

E. Qualtative Analysis

To qualitatively analyze and demonstrate the effectiveness of employing narration in text-video retrieval, we provide additional examples of retrieval results and generated frame-level captions for three datasets: Fig. 11 for MSVD, Fig. 12 for VATEX, and Fig. 13 for DiDeMo. Additionally, Fig. 14 shows examples with incorrect results due to short and general queries.

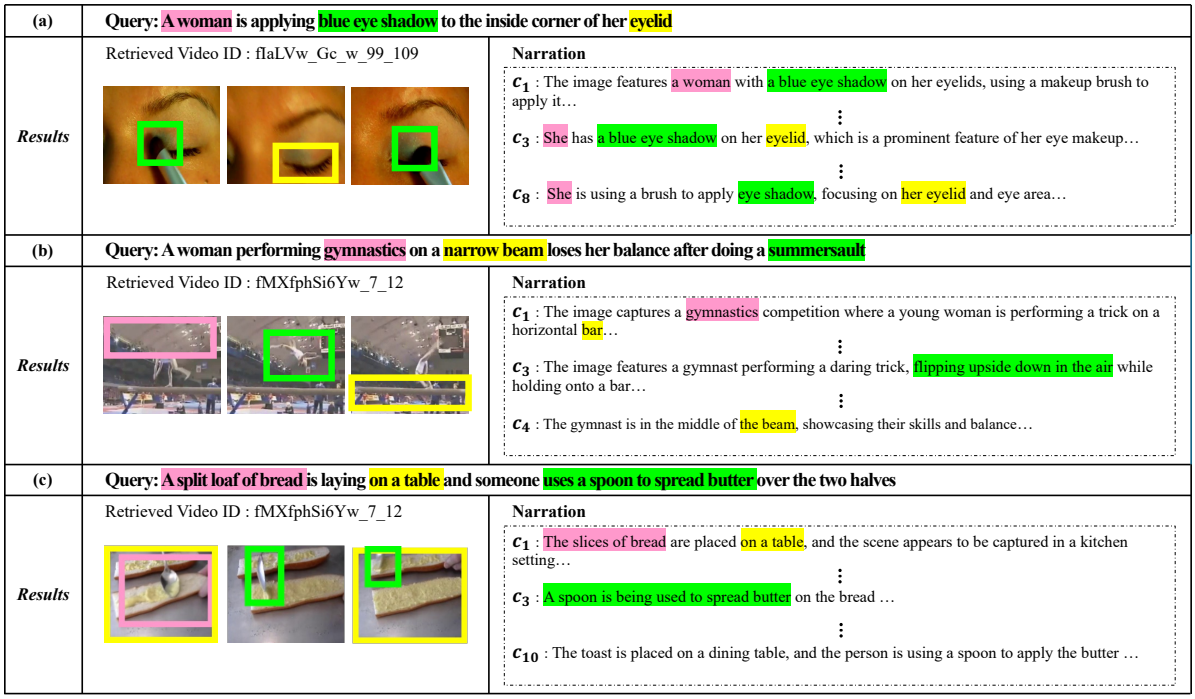

Figure 11. Text-to-video retrieval results on MSVD. (a) In the provided query, NarVid effectively identified the subject, a woman, and accurately recognized details such as the blue eye shadow color and the makeup situation on the eyelid. (b) NarVid demonstrates a good understanding of gymnastics, specifically the narrow beam and the somersault action, describing it as flipping upside down in detail. (c) NarVid not only understands the layout of the bread and table but also detects the spoon's movement to locate the right video.

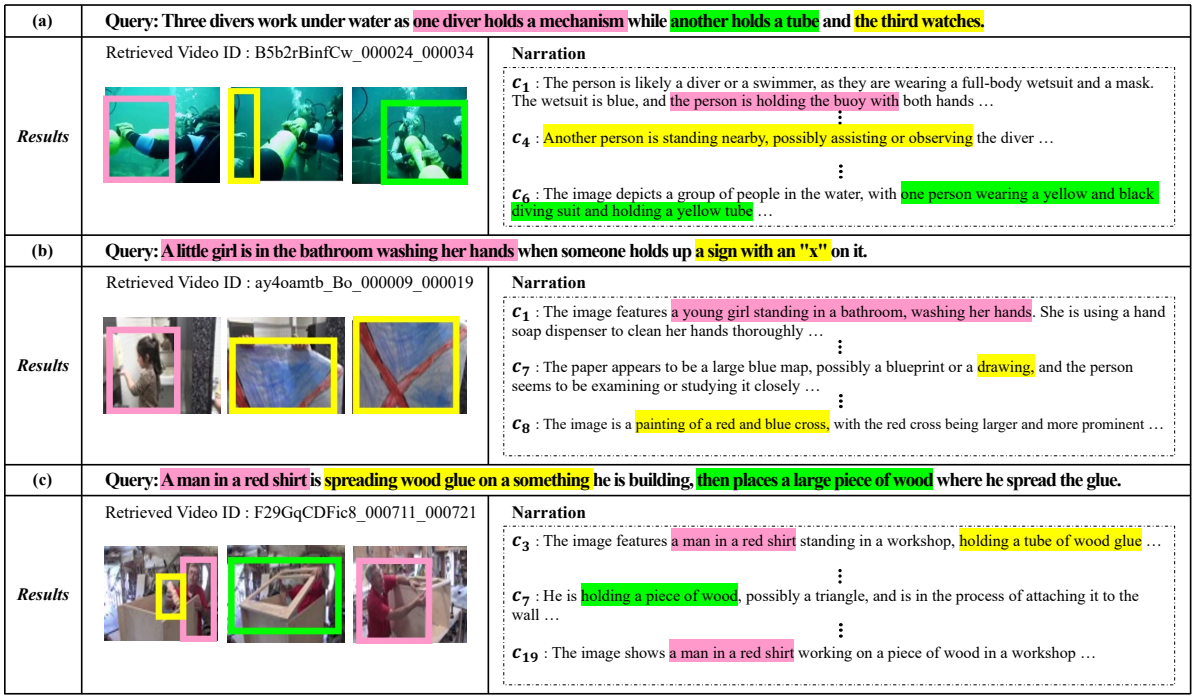

Figure 12. Text-to-video retrieval results on VATEX. (a) The narration effectively captures the actions of the three individuals in the video, leading to positive retrieval results for Narvid. (b) The narration effectively identifies local information in two scenes: One showing the girl washing her hands and another with her holding up the "x" sign. (c) Narvid correctly identifies the man in a red shirt by observing woodworking behaviors like using wood glue and handling wood pieces over time.

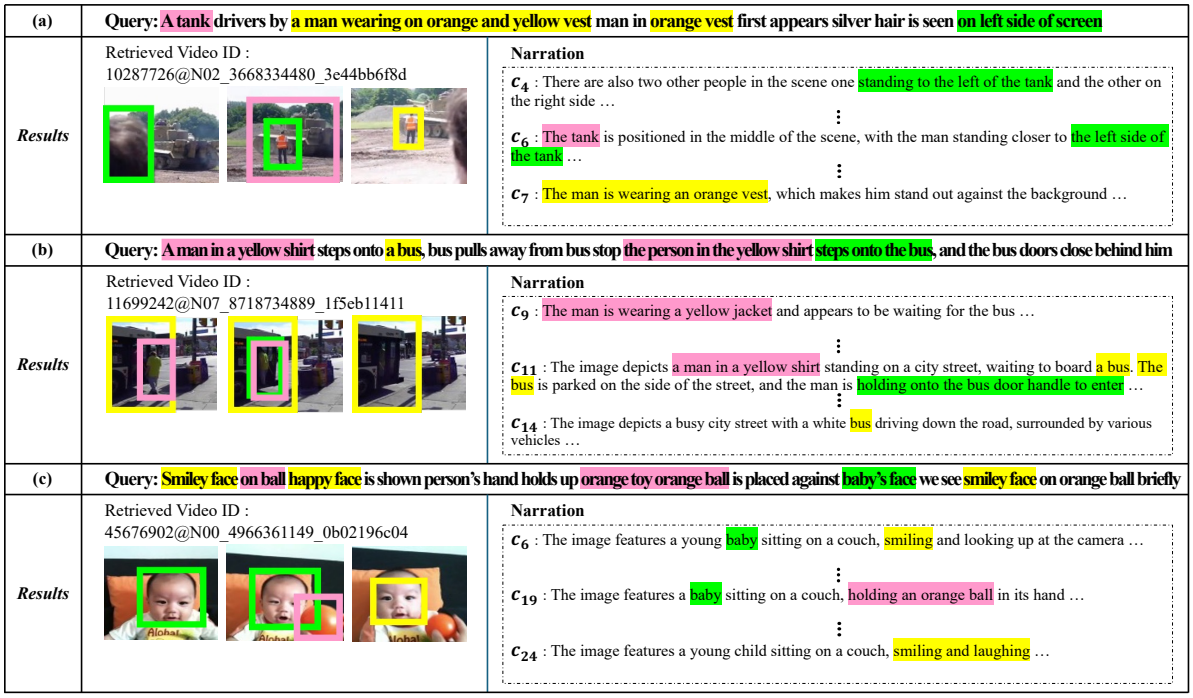

Figure 13. Text-to-video retrieval results on DiDeMo. (a) The narration is effective in identifying objects like a tank or a man in an orange vest in videos, as well as indicating location expressions like left. (b) NarVid clearly understands the behavior of the man wearing a yellow shirt over time and notes that he boards the bus. (c) The narration assists in identifying the baby, the ball, and the baby's joyful expression in the video.

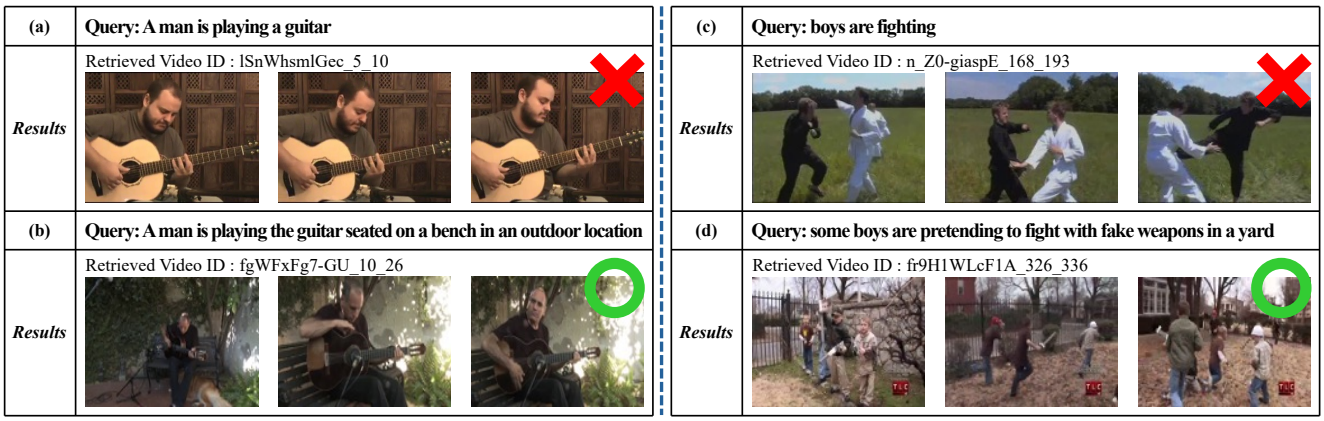

Figure 14. Example of mismatched results from NarVid due to a short query in the MSVD dataset: Queries (a) and (b) demonstrate how a short query with less information can lead to similar but incorrect retrieval results. In contrast, a detailed query specifying the location (sitting on a bench and outdoors) yields the correct answer. Similarly, queries (c) and (d) illustrate that adding specific details, such as holding weapons or fighting in the yard, to the short query (boys are fighting) can produce an accurate result.