Appendix for
GEXIA: Granularity Expansion and Iterative Approximation for Scalable Multi-grained Video-language Learning

A. Ablation Studies

Encoder Weight Setup: We perform an ablation study to explore different weight setups for the dense feature encoders prior to pretraining and present the finetuning retrieval results on the ActivityNet Captions dataset in Table 1. These results demonstrate a significant improvement when initializing the feature backbone with CLIP pretraining weights as opposed to training from scratch. It's evident that the InternVid-10M-FLT dataset isn't large enough to independently train our model from scratch. The common practice of initializing the video feature backbone with pretrained weights from an image model [16] works well for large-scale video-text pretraining. Furthermore, our observations indicate that freezing the encoders leads to inferior outcomes. The best way is to unfreeze the backbone with a smaller learning rate. This strategy suggests that the image representations learned from CLIP do not seamlessly transfer to video representations.

Random Integration in GEX: We conduct an ablation experiment to further demonstrate that, even for single-grained pretraining datasets without prior information, using random GEX concatenation operations ($\oplus_v$ and $\oplus_t$) can still improve the model's performance. We replace Long-Video-Long-Text (LVLT) and Long-Video-Short-Text (LVST) pairs with the same number of randomly concatenated short video-text pairs. The rest of the setups remain consistent. As the results of T2V and V2T retrieval on ActivityNet Captions shown in Table 2, random concatenation is suboptimal to the current data pipeline, leading to 0.5%/1.1% (T2V/V2T R@1) performance drop. Compared to the approach with SVST only, random concatenation still provides 3.2%/2.7% performance improvements, which can be viewed as an alternative way when the prior knowledge of videos is not available.

LLM for Text Compression (Summarization) in GEX: We test the effectiveness of various open-source Large Language Models (LLMs) serving as the Text Compression (Θ_t) operation to create summaries. This assessment is carried out through 100 summarization tasks, randomly selected from the LSMDC validation set, which comprises 362 short video clips. Using GPT4's summaries as references, we evaluate different models' performance based on the relevance score of their generated summaries, involving ROUGE score [12] and BERTScore [25], along with the average running time on one summarization instance as presented in Table 3. Vicuna 13b-v1.5 [5] emerges as the top performer, particularly in terms of the highest BERTScore, striking the best balance between performance and runtime. Note that we don't use GPT4 directly as the Text Compression operator due to its high cost and limitations in parallel processing capabilities.

Table 1. Retrieval results of finetuned model on ActivityNet Captions dataset across different encoder weight setups.

Encoder Weight Setup	Text-to-video			Video-to-text		
	R@1	R@5	R@10	R@1	R@5	R@10
Scratch	24.7	51.1	64.1	24.7	52.6	65.9
Freeze CLIP	37.5	67.4	80.6	36.9	68.0	80.9
Unfreeze CLIP	**45.3**	**76.5**	**86.6**	**45.0**	**76.4**	**87.3**

Table 2. Retrieval results of finetuned model on ActivityNet Captions dataset across different concatenation operations in GEX.

Pretraining Data Pairs	Text-to-video			Video-to-text		
	R@1	R@5	R@10	R@1	R@5	R@10
SVST only	41.6	72.8	83.7	41.2	72.6	84.0
Random Concat.	44.8	75.1	86.1	43.9	74.7	85.5
Concat. w/ source IDs	**45.3**	**76.5**	**86.6**	**45.0**	**76.4**	**87.3**

B. Semantic Alignment between Long and Summarized Texts

To verify that the summarized short texts retain semantic similarity with the original concatenated long texts, ensuring consistency between them, we analyze the cosine similarity between the long and summarized short text features.

We randomly sample 100 concatenated long videos along with their corresponding long texts and summarized short texts, and compute their CLIP-based [16] features. Next, we calculate the average cosine similarities between

Table 3. LLM assessment results. RG: ROUGE score [12]; BERT: BERTScore-F1 [25]; Time: Average running time on one summarization task. The model underlined is the final selected LLM.

LLM	RG-1	RG-2	RG-L	BERT	Time (s)	LLM	RG-1	RG-2	RG-L	BERT	Time (s)
Longchat 7b [11]	0.43	0.20	0.34	0.90	2.45	Internlm 7b [18]	0.32	0.13	0.26	0.87	3.36
OpenLlama 7b [9]	0.48	0.24	0.39	0.89	1.03	RWKV-4 7b [3]	0.28	0.11	0.23	0.88	1.04
OpenLlama 13b [9]	0.38	0.18	0.31	0.90	2.19	Vicuna 7b [5]	0.46	0.23	0.37	0.91	1.91
Fastchat t5 3b [28]	0.48	0.25	0.40	0.91	2.97	Vicuna 13b [5]	0.48	0.24	0.39	0.92	2.19
Dolly-v2-7b [6]	0.36	0.15	0.28	0.89	4.93	Vicuna 33b [5]	0.50	0.27	0.42	0.92	7.38

Table 4. The average cosine similarities of CLIP-based features between concatenated long videos, concatenated long texts, and summarized short texts in the pretrained dataset.

Features	concat. long videos concat. long texts (100 samples)	concat. long videos sum. short texts (100 samples)	concat. long texts sum. short texts (1M)
Avg±Std Cos Sim	0.220±0.037	0.215±0.033	0.791±0.084

the full set of 1M text features and the 100 sampled video-text feature pairs, as shown in Table 4. The t-SNE visualization of the 100 sampled features is also provided in Figure 1, where we observe the clustering of the features.

From the similarities in Table 4, we observe that the long-video-long-text pairs (LVLT) and long-video-short-text pairs (LVST) exhibit similar levels of similarity in their CLIP-based features (0.220±0.037 vs. 0.215±0.033, respectively). The relatively low similarity between video and text features can be attributed to the domain gap between the image-pretraining dataset of the CLIP model and this video dataset. However, both the t-SNE plot and the cosine similarity scores show a strong resemblance between the summarized short texts and the original concatenated long texts, reflected in the much higher similarity score of 0.791. These findings suggest that the summarized short texts preserve enough semantic information from the long texts to effectively serve as positive samples with the long videos for subsequent alignment learning.

C. Retrieval Complexity and Inference Efficiency

Our GEXIA method incorporates separate video and text branches, similar in structure to CLIP4Clip [13], maintaining the same efficient retrieval complexity of $\mathcal{O}(N_v N_t)$, where N_v represents the number of candidate videos and N_t is the size of the text query set. The video and text IAMs are connected with the video and text branches respectively, thus the $\#iter$ setups do not affect the retrieval complexity. In contrast, retrieval-specific methods like X-CLIP [14] introduce cross-modal and fine-grained features before retrieval, resulting in a much higher complexity of $\mathcal{O}(N_v N_t N_f N_w)$, where N_f denotes the number of frames per video, and N_w represents the average length of words

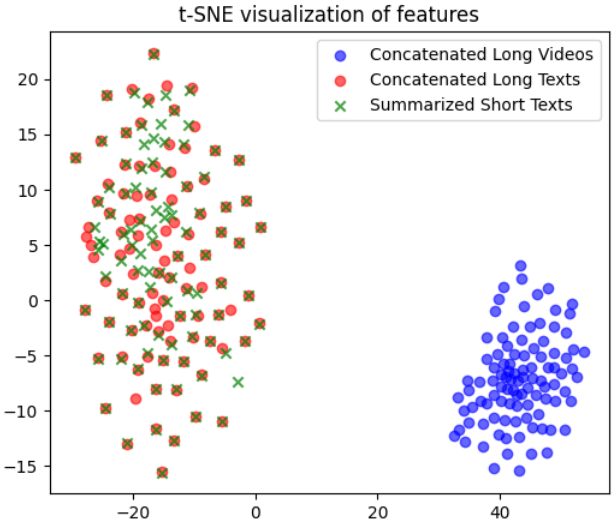

Figure 1. t-SNE visualization of the CLIP-based features for the sampled 100 concatenated long videos, concatenated long texts, and summarized short texts.

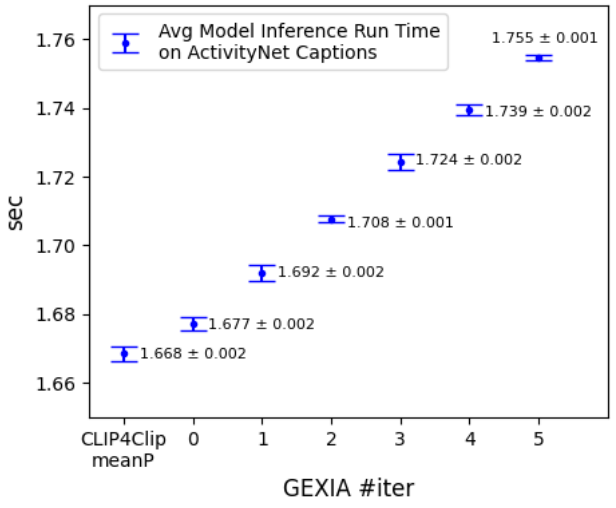

Figure 2. Average model inference run time on ActivityNet Captions across different $\#iter$ setups for GEXIA, compared to CLIP4Clip (mean pooling setup).

in the texts. Additionally, many retrieval-specific models [2, 7, 14, 21, 27] are finetuned exclusively on retrieval datasets, which limits their applicability as foundational models for broader video understanding and classification tasks.

To further demonstrate the efficiency of our model, we present the average inference run time across different $\#iter$ setups, compared to CLIP4Clip (mean pooling setup) [13]. The experiment was conducted on one NVIDIA Tesla-V100 GPU with a batch size of 64 using the ActivityNet Captions dataset. As shown in Figure 2, our model's run time is only slightly longer than CLIP4Clip, due to the smaller feature size and minimal computational overhead introduced by the iterative approximation process. Specifically, with $\#iter$=3, our model achieves an 11.8% relative improvement in text-to-video R@1 on the ActivityNet Captions dataset (according to Table 1 in the paper), while requiring only 3.3% more inference time compared to CLIP4Clip. These results highlight the efficiency of our model, offering substantial performance gains with only a modest increase in computational cost.

D. Comparison of Computational Costs

We compare the computational costs between our model and various baseline models, using FLOPs (Floating Point Operations) per frame as the metric. As illustrated in Table 5, our model demonstrates one of the lowest FLOPs among the compared models, largely due to the use of a compact backbone (ViT-B/32) for the local video encoder. Notably, the ViCLIP model incurs a computational cost that is 17× higher than our model. This lower computational cost makes our model not only efficient in terms of inference time but also in terms of overall computational resource consumption, enhancing its applicability in real-world scenarios.

E. Detailed Dataset Statistics and $\#iter$ Setups

We provide the detailed average video/text lengths of each downstream dataset and the corresponding $\#iter$ setups in Table 6. We set these $\#iter$s based on the average length of the videos and texts of the given datasets, where $\#iter = 3$ for long video/text data and $\#iter = 1$ for short ones.

F. Extension to a New Granularity: Image-Text Data

To further explore the generalization capability of our method, we extended its application to include a new type of data granularity: images to short texts, where images can be seen as one-frame videos. This expansion involves generating image-text pairs for pretraining, achieved by the Video Compression (Θ_v). Here we extract the middle frame from each short video in the InternVid-10M-FLT [20] dataset and pair it with the corresponding text of the video. As such, we compose 10M image-text pairs. Following this, we take the model that was initially pretrained on video-text pairs and proceed with further pretraining using these newly formed image-text pairs for 5 epochs. Additionally, we set $\#iter = 0$ for the video branch to differentiate the granularities of the image and video and keep the text $\#iter$ as 1.

After completion of the pre-training phase, we use the zero-shot ImageNet [8] classification task as a benchmark to assess the generalizability of our GEXIA method. For this purpose, to implement the pretrained model for zero-shot image classification, following the CLIP setup [16], we employ a prompt template: "A video of a {label}." to transform the problem into zero-shot image-text retrieval. The results of the experiment are shown in Table 7. Our model demonstrates remarkable generalization capability to the new granularity of image-short-text, surpassing reference

Table 5. GFLOPs per frame of our model and other baselines.

Model	Vision Backbone	GFLOPs
CLIPBert [10]	Res_{50}	4.1
TACo [22]	Res_{152}	11.6
Frozen [2]	TS4mer	44.5
LF-VILA [17]	Swin_B	9.3
HiTeA [24]	MViT_B	8.2
LocVTP [4]	ViT_{B16}	17.6
ViCLIP [20]	ViT_{L14}	81.1
VideoPrism [26]	ViViT_B	$\geq$ 142.0 [1]
Ours	ViT_{B32}	4.7

Table 6. Average video/text lengths of the seven downstream datasets and the corresponding $\#iter$ setups.

Dataset	Video (sec)	Text (#word)	$\#iter$ (V-T)
ActivityNet	180	49.2	3-3
MSR-VTT	14.8	9.3	1-1
LSMDC	4.7	9.7	1-1
LVU	120	N/A	3 (only V)
COIN	141.6	N/A	3 (only V)
Charades-Ego	31.5	3.9	3-1
How2QA	17.5	16.0	1-1

Table 7. Zero-shot ImageNet [8] classification accuracy results. All models utilize a ViT-B/32 backbone. ZS: Zero-shot; IT: Image-Text data pairs; VT: Video-Text data pairs; $\#iter$: Video-Text iteration number.

Method	Pretraining Dataset	#PT Granularities	ZS Acc.
CLIP [16]	YFCC-15M [19]	1 (IT only)	32.8
SLIP [15]	YFCC-15M [19]	1 (IT only)	34.3
FILIP [23]	YFCC-15M [19]	1 (IT only)	39.5
Ours ($\#iter$: 1-1)	InternVid-10M [20]	3 (VT only)	32.5
Ours ($\#iter$: 3-1)	InternVid-10M [20]	3 (VT only)	31.0
Ours ($\#iter$: 1-1)	InternVid-10M [20]	4 (VT+IT)	33.9
Ours ($\#iter$: 3-1)	InternVid-10M [20]	4 (VT+IT)	31.5
Ours ($\#iter$: 0-1)	InternVid-10M [20]	4 (VT+IT)	**40.6**

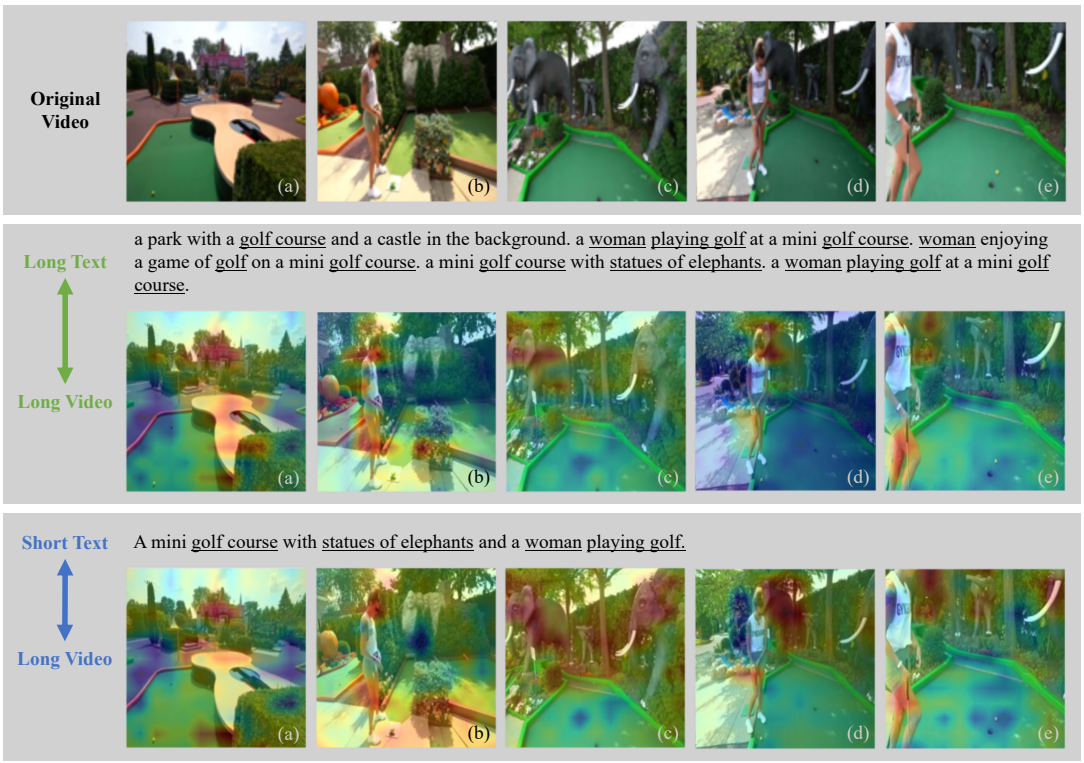

Figure 3. The visualization of alignment scores for long-video-long-text and long-video-short-text pairs. Given the same video, our GEXIA method is able to capture and align different information according to the input texts. Areas highlighted in red indicate regions of higher alignment scores, whereas the blue regions represent areas with lower alignment scores.

Figure 4. The visualization of alignment scores for different $\#iter$ settings. The 3-3 setup of $\#iter$ works better than the 1-1 in this long-video-long-text case. Areas highlighted in red indicate regions of higher alignment scores, whereas the blue regions represent areas with lower alignment scores.

models targeting image tasks by a large margin. The comparison across our five models shows that setting $\#iter = 0$ for image inputs and incorporating image-text pairs into the pretraining data leads to the highest performance for image-based tasks. This finding further affirms that assigning appropriate iteration numbers based on the granularity of the input, along with pretraining data of the targeted granularity, is the key to achieving effective multi-grained visual-language alignments.

G. Qualitative Study of Cross-modal Alignments

We qualitatively study the video-text alignments of our GEXIA models by visualizing pixel-level alignment scores across temporal and spatial dimensions. Given an input video v and text t, we start by computing the similarity value S between the output video and text embeddings. Next, we create a modified version of the video by masking a small patch at coordinates $[h, w]$ in one frame t of the video, resulting in $v^{\text{mask}}_{t,h,w}$. We then calculate the similarity value $S^{\text{mask}}_{t,h,w}$ between the embeddings of the masked video and the text.

The difference between $S_{t,h,w}$ and $S^{\text{mask}}_{t,h,w}$ is defined as the alignment score at the patch level, indicating the reduction in alignment when the mask is applied. We apply a 32×32 mask patch across the video frames in a sliding window fashion with a 16-pixel stride, producing a 13×13 matrix of patch-level alignment scores for each frame. We then resize these scores to match the original input video dimensions, resulting in the pixel-level alignment scores. We visualize the pixel-level scores in Figure 3 and Figure 4.

G.1. Alignments with Long and Short Texts

In Figure 3, when presented with the same long video, we observe notable differences between long text and short text inputs. For the long text, which contains more detailed information, the alignment scores are dispersed more uniformly across various regions and objects within the video. This indicates a comprehensive integration of video content with extensive textual details. Conversely, in the case of short text, the alignment scores are more focused on specific key elements mentioned in the text. For example, the model concentrates on "the status of elephants" in frames (c) and (e), and on "a woman playing golf" in frame (b). This pattern reveals that our model is adept at aligning video content with both long- and short-text inputs, effectively adjusting its focus based on the granularity of the text.

G.2. Alignments with Different Iteration Numbers

In Figure 4, we further study the impact of the iteration number $\#iter$ in a qualitative way. This assessment involves a comparison of visualized pixel-level alignment scores using two $\#iter$ settings for a given pair of long video and long text. The figure reveals that when $\#iter$ is set to 1-1, the model struggles to identify key details in the text, notably missing elements like "scooter" and "rake" in frames (b), (c), and (d). Additionally, the alignment scores appear more randomly scattered across the frames, suggesting suboptimal alignment in this configuration. On the other hand, the $\#iter$ set to 3-3 shows a contrast. This setup enables the model to detect all critical details in the text, reflected in high alignment scores for corresponding objects in the video. This difference in performance between the two $\#iter$ settings not only highlights the significant role of iteration numbers during inference but also reaffirms the adaptability of our model to multi-grained video-text pairs.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 3

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 3

[3] PENG Bo. Blinkdl/rwkv-lm. https://doi.org/10.5281/zenodo.5196577, Aug. 2021. 2

[4] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. In *European Conference on Computer Vision*, 2022. 3

[5] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/, March 2023. 1, 2

[6] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm, 2023. 2

[7] Chaorui Deng, Qi Chen, Pengda Qin, Da Chen, and Qi Wu. Prompt switch: Efficient clip adaptation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15648–15658, 2023. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[9] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama. https://github.com/openlm-research/open_llama. 2

[10] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341, 2021. 3

[11] Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can open-source llms truly promise on context length? https://lmsys.org/blog/2023-06-29-longchat, June 2023. 2

[12] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 1, 2

[13] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2, 3

[14] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 2, 3

[15] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022. 3

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3

[17] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *Advances in neural information processing systems*, 35:38032–38045, 2022. 3

[18] InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM, 2023. 2

[19] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 3

[20] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 3

[21] Ziyang Wang, Yi-Lin Sung, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Unified coarse-to-fine alignment for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2816–2827, 2023. 3

[22] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11562–11572, 2021. 3

[23] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2021. 3

[24] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15405–15416, 2023. 3

[25] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. 1, 2

[26] Long Zhao, Nitesh Bharadwaj Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. In *Forty-first International Conference on Machine Learning*. 3

[27] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 970–981, 2022. 3

[28] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 2