ReSpec: Relevance and Specificity Grounded Online Filtering for Learning on Video-Text Data Streams

Supplementary Material

1. Appendix: Table of Contents

The Appendix enlists the following additional materials.

2. Extended Related Works

2.1. Robust Learning on Noisy Dataset

Noisy correspondence learning focuses on research related to mismatched pairs in multimodal data. Its goal is to train models robustly in noisy correspondences, generally by measuring the degree of alignment between modalities and reflecting it in the training process. In early research (NCR [8]), the clean/noisy data is classified using a GMM-based method by utilizing the loss difference between clean and noisy data, which stems from co-training and memorization effects. NPC [31] measures the correspondence based on the performance difference between the current data and clean data trained on a similar sample.

This approach has recently been extended to video-text datasets, where temporal misalignment is considered. In such cases, frame-text alignment is measured using Optimal Transport (NorTon [11]) or mutual agreement-based methods (TempCLR [28]), which are then reflected in the learning process. Noisy correspondence learning generally involve storing the entire dataset and conducting several iterations/epochs to train a generalizable model from the noisy dataset, incurring significant storage and computational costs.

2.2. Online Learning

Online learning has largely been explored in the context of continual learning, which studies the problem of learning sequential tasks while alleviating catastrophic forgetting on the past tasks [1, 5, 15, 24]. OCL-NDS [24] proposes adaptive learning rate scheduling and replay buffer size adaptation algorithm for online continual learning with natural distribution shift. CVT [5] proposes an attention-based mechanism to mitigate the catastrophic forgetting for online continual contrastive learning.

Online learning has also been used to improve the training efficiency by specializing a model on specific target tasks [3, 7, 17]. For instance, OMD [17] trains a small network on a long video stream using online model distillation from high capacity teacher model, obtaining specialized model that performs on par with the much larger teacher on the target video stream.

3. Experiment Details

3.1. Dataset Details

The input data stream we consider consists of two web-scale noisy video-text datasets. First, VideoCC3M [18] is a large-scale, web-curated dataset containing approximately 2.5 million video-text pairs. It is constructed by mining videos that are similar to seed images from CC3M [22] and transferring the images' captions to the videos. Since both the construction of CC3M and the video mining of VideoCC3M are done automatically, the videos and texts in VideoCC3M are considered weakly paired and contains various types and degrees of noise.

Another noisy source dataset we use is WebVid2M [2], consisting of approximately 2.5 million clips and corresponding captions scraped from the stock footage sites. While WebVid2M is considered more *clean* than VideoCC3M since the captions are human-generated, it still contains noisy correspondences between video and text, such as text containing the metadata of the video rather than its description.

3.2. Baseline Details

Cosine similarity thresholding. When constructing a large-scale multimodal datasets, it has become a common practice to compute CLIP [19] cosine similarity of image and text embeddings and drop samples below a certain threshold to filter out unsuitable image-text pairs. For instance, the threshold of 0.3 is used in LAION-400M [20]

and the thresholds of 0.28 and 0.26 are used in LAION-5B [21] depending on the language of the text. Similar approach is also used to filter video-text datasets, as Koala [23] computes the frame-wise CLIP cosine similarity and only use samples whose max frame-wise cosine similarity is greater than or equal to 0.26 as training data. Thus as baselines, we compute frame-wise image-text CLIP cosine similarity and apply threshold values of $0.3, 0.28, 0.26, 0.24, 0.22$, and 0.2 on the cosine similarities. We use three variants of CLIP Threshold, each using the average, max, and middle-frame cosine similarity. Similarly, we also compute video-text cosine similarity between text and video using LanguageBind [32] and apply the threshold values on. Note that these baselines only consider the multimodal alignment and do not account for the downstream task relevance and specificity.

Online downstream task-aware filtering. We adapt previous online downstream task-aware filtering approaches CiT [26] and CoLoR-Filter [4] to our setting.

While CiT [26] is originally designed for training a single joint filtering and training network, we adapt it to our setting by using separate filtering (LanguageBind) and training (BT-Adapter) networks for fair comparison. CiT uses cosine similarity between text of incoming data and texts from downstream task data as a measure of relevance and selects sample whose cosine similarity to downstream task data is larger than a certain threshold. The selected samples are also used for the training of the filtering network. For computational efficiency, we only update the parameters of the projection layers. Following the original work, we use AdamW [13] optimizer with learning rate of 5e-4 and weight decay of 1.0, and batch size of 1,536 (CiT single GPU setting) for the filtering network parameter update. We use the text cosine similarity threshold of 0.55. Section 5.3 provide ablation on the text cosine similarity threshold.

CoLoR-Filter [4] is originally designed to select downstream relevant data for language models by finetuning the prior language model on downstream dataset and using the difference of language modeling loss (negative log probability) of the data computed using the finetuned model and the prior model as the criterion. We extend it to online sample-wise video-text data filtering by using LanguageBind as the prior model and finetuning it on downstream datasets via contrastive loss. For copmutational efficiency, we finetune the projection layers of LanguageBind using the downstream datasets for ten epochs using AdamW optimizer with learning rate of 1e-3 and weight decay of 0.2, and batch size of 4096.

While the original CoLoR-Filter ranks samples within a mini-batch and select certain proportion of the data based on the ranking, we modify it to work in online, sample-wise setting. We use the difference of the video-text cosine similarity of the data between the finetuned and the prior model as the criterion, and only sample when the cosine similarity measured using the finetuned model is larger. We also provide the result of using the original ranking-based selection using mini-batch of data in Section 5.3.

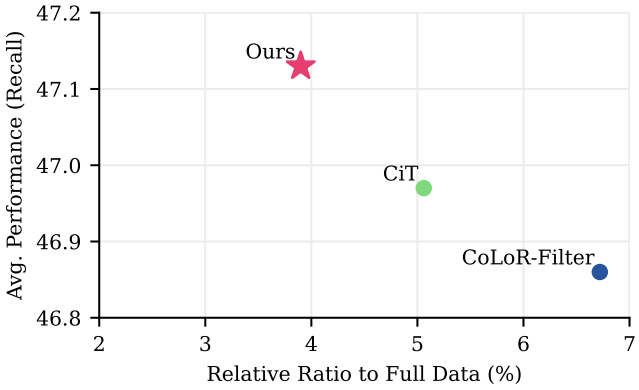


Figure 1. **Performance comparison on HowTo10M.** We compare our approach to the top performing baselines based on the average performance and the ratio of filtered data size to full data size (HowTo10M here). The average performance is the average of Recall at 1, 5 and 10 across five downstream tasks.

Since these two approaches do not consider the multimodal alignment between video and text (*i.e.*, cleanness), we use LanguageBind video-text cosine similarity thresholding before applying them for fair comparison. We report the best result across the aforementioned video-text cosine similarity threshold values.

3.3. Training Details

For ReSpec and other baselines, we use the filtered data for the online training of BT-Adapter [12]. We use BT-Adapter with OpenAI CLIP-L/14 backbone, and follow the implementation details of the original work [12] for the masking ratio, temperature scale, and the number of adapted layers. We use AdamW [13] optimizer with learning rate 2e-6 and weight decay 0.05 as in the original implementaion, with batch size of 52 for WebVid2M and 100 for VideoCC3M.

4. Extended Results

4.1. Additional Dataset Results

We conduct additional experiments on the HowTo10M dataset, a subset of HowTo100M [16] comprising approximately 10% of the original data. Unlike the previously used datasets, HowTo10M is a web-crawled video-text dataset relying on Automatic Speech Recognition (ASR) for video-text alignment. This experiment additionally investigates whether our model maintains its performance on an ASR-based video-text dataset.

Model	Clip ratio (%)	MSR-VTT			DiDeMo			ActivityNet			YouCook2			LSMDC			Avg. Perf.
		R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	
Full data	100.00%	39.70	64.70	72.80	34.52	60.02	69.64	38.45	68.52	80.42	11.07	26.29	36.22	20.70	39.30	46.10	47.23
CLIP Avg Threshold	94.51%	41.50	64.60	73.80	35.62	59.42	70.24	38.89	68.01	80.17	10.70	27.26	37.19	22.88	40.36	46.25	47.79
CLIP Mid Threshold	86.11%	40.60	64.50	73.00	35.42	61.31	70.44	39.23	68.79	80.49	10.70	26.70	36.43	23.20	40.40	46.60	47.85
CLIP Max Threshold	98.11%	41.30	63.70	72.50	35.42	60.71	69.94	38.67	68.50	80.35	10.79	27.50	36.72	22.00	41.50	47.40	47.80
LB Threshold	86.32%	40.20	64.80	73.30	36.08	61.15	70.76	40.17	69.52	81.25	10.88	27.25	37.03	22.50	40.50	47.10	48.17
CiT [26]	42.53%	42.00	66.00	75.60	35.84	61.78	70.79	40.57	70.15	82.09	10.76	27.96	37.38	24.00	41.20	48.70	48.99
CoLoR-Filter [4]	56.36%	41.70	65.50	73.30	34.82	61.81	71.33	40.50	69.21	80.82	11.02	27.56	37.24	23.40	40.90	47.80	48.46
ReSpec (ours)	27.50%	42.10	67.00	76.10	36.31	62.30	72.42	40.50	69.79	81.69	11.33	26.75	37.20	24.10	40.60	48.50	49.11

Table 1. **Performance on 5 downstream tasks trained with WebVid2M dataset.**

Model	Clip ratio (%)	MSR-VTT			DiDeMo			ActivityNet			YouCook2			LSMDC			Avg. Perf.
		R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	
Full data	100.00%	35.00	58.80	69.80	34.79	58.97	70.27	31.70	60.48	74.61	6.86	18.33	25.62	18.90	34.10	41.10	42.62
CLIP Avg Threshold	15.57%	39.60	64.70	73.70	34.72	60.02	68.65	35.34	65.10	77.56	9.59	23.76	33.17	20.90	38.40	46.00	46.08
CLIP Mid Threshold	43.62%	39.90	63.50	73.60	35.32	59.72	70.04	34.87	62.83	76.38	9.24	23.18	32.54	22.10	38.10	44.20	45.70
CLIP Max Threshold	39.36%	38.30	64.60	74.40	34.79	59.86	70.37	35.37	63.67	76.58	9.07	23.55	32.57	21.70	38.80	45.50	45.94
LB Threshold	23.95%	40.00	65.90	75.40	35.74	60.79	70.89	37.33	66.76	79.06	9.99	24.24	33.43	22.48	39.36	46.85	47.21
CiT	20.69%	40.60	65.90	75.70	35.64	62.18	70.89	37.78	65.77	78.37	10.01	24.89	33.10	21.98	38.96	46.55	47.22
CoLoR-Filter	12.51%	41.30	65.10	76.10	35.45	60.40	70.00	37.87	66.58	79.44	10.10	24.96	34.23	22.70	38.50	46.10	47.25
ReSpec (ours)	5.41%	40.80	65.50	75.00	34.99	61.45	70.17	37.28	66.88	79.78	10.27	25.83	35.01	21.68	39.06	46.45	47.34

Table 2. **Performance on 5 downstream tasks trained with VideoCC3M dataset.**

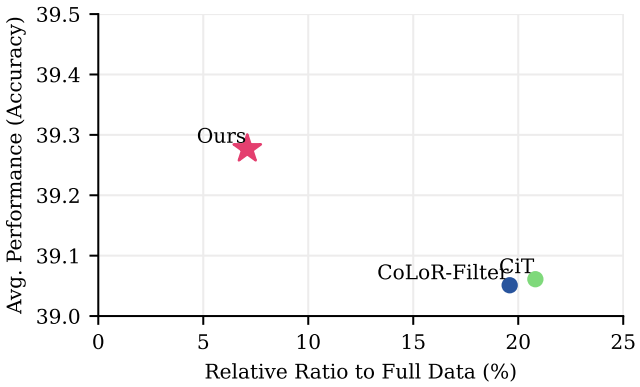


Figure 2. **Additional architecture performance comparison on VideoCC3M** We compare our approach using the FrozenBiLM architecture [27] to the top performing baselines based on the average performance and the ratio of filtered data size to full data size (VideoCC3M here). The average performance is the average of Recall at 1, 5 and 10 across four downstream tasks.

As shown in Fig. 1, ReSpec delivers the best performance, requiring the least amount of data while achieving the highest average performance across five downstream tasks.

4.2. Additional Architecture and Downstream Tasks Results

In addition to BT-Adapter [12], we also validate the efficacy of ReSpec on another architecture, FrozenBiLM [27], which is also well-suited for online training. FrozenBiLM endows the pretrained bi-directional language model with zero-shot video question answering capabilities by freezing the visual backbone and the bi-directional language model and training visual-text projection and adapter layers using the masked language modeling (MLM) objective.

Unlike BT-Adapter, which used zero-shot video-text retrieval as downstream tasks, we use four zero-shot open-ended video question-answering tasks (MSRVTT-QA [25], MSVD-QA [25], ActivityNet-QA [30], and TGIF-QA [9]) and one zero-shot video-conditioned fill-in-the-blank task (LSMDC-FIB [14]) as downstream tasks of FrozenBiLM.

Fig. 2 reports the result of the online trained FrozenBiLM on video-text data filtered from VideoCC3M. ReSpec outperforms CiT and CoLoR-Filter in terms of average performance (average of top-1 and top-10 accuracy) while being most efficient in terms of the amount of the selected data.

4.3. Generalization to Online Image-Text Filtering

While we mainly focus on the *video*-text domain since online filtering is more critical for video data, where storage and computational demands are significantly higher, ReSpec is generalizable to online image-text filtering. To

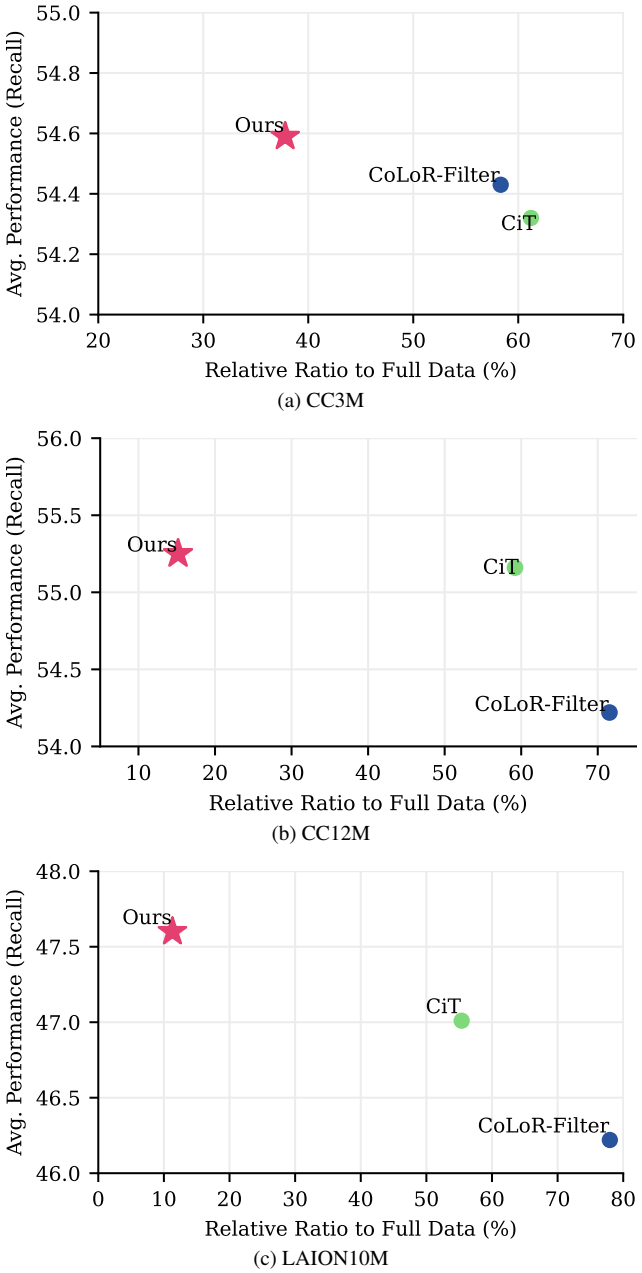


Figure 3. **Performance comparison on online image-text filtering** We compare our approach to the baselines based on two metrics: average performance and the ratio of filtered data size to full data size. The evaluation is conducted on three datasets: CC3M [22], CC12M [6], and a 10M subset of LAION-400M [20]. The average performance is computed as the mean of Recall at 1, 5, and 10 across two downstream tasks.

demonstrate the generalizability, we conduct experiments on online image-text filtering and training by using CLIP ViT-B/16 features for filtering and training CiT [26] architecture with ViT-B/16 image encoder pretrained on ImageNet21k and pretrained SimCSE-BERT$_{\text{base}}$ text encoder using the filtered image-text pairs in an online manner. we freeze the pretrained image encoder and train the text encoder and two projection layers. we use COCO [10] and Flickr30k [29] retrieval as the downstream tasks.

Fig. 3 compares ReSpec to CiT and CoLoR-Filter when using CC3M [22], CC12M [6], and 10M subset of Laion-400M [20] as the noisy source datasets. in all three settings, ReSpec achieves the best average performance (average of Recall at 1, 5, and 10) while using the smallest amount of data.

4.4. Multi-Dataset Training Results

We conduct two multi-dataset training experiments: one using a sequential stream from VideoCC3M to WebVid2M, and the other with a randomly shuffled stream of both datasets. We evaluate performance using two metrics: average performance (mean Recall at 1, 5, and 10 across five downstream tasks) and the relative ratio of filtered data size to total data size.

First, Fig. 4 compares sequential multi-dataset training (VideoCC3M $\rightarrow$ WebVid2M) across different approaches. Our method outperforms CiT in average performance and requires significantly less data, with a ratio of 21%, compared to CiT's 31%. In contrast, CoLoR-Filter shows slightly lower performance and a data usage ratio similar to CiT's, while the LB Threshold method requires the largest data volume (55%) but performs the worst. This highlights our approach's efficiency in minimizing data requirements while maintaining top-tier recall performance across downstream tasks with sequential multi-dataset training.

Second, Fig. 5 shows results from training on a randomly shuffled stream of VideoCC3M and WebVid2M. This experiment tests whether the model retains robust performance despite the random dataset order. Our approach again outperforms others, requiring only 16% of the data, compared to CiT's 32% and CoLoR-Filter's 34%. This further underscores the efficiency of our method in minimizing data usage while maintaining state-of-the-art recall performance across downstream tasks, even with randomly shuffled multi-datasets.

4.5. Per-Task Results

Tab. 1–2 shows the recall at 1, 5, and 10 for each downstream tasks and the average of the recalls across five downstream tasks (Avg. Perf.), along with the clip ratio (the ratio of the filtered data size over the original dataset size). ReSpec achieves the best average performance while filtering the least amount of data in both WebVid2M and VideoCC3M.

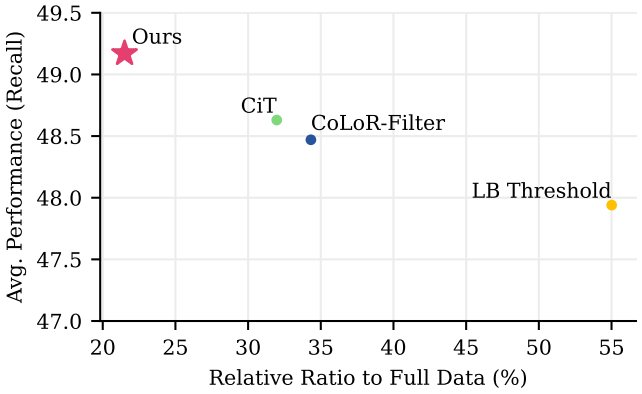


Figure 4. **Online multi-dataset (VideoCC3M → WebVid2M) filtering and training performance comparison.** We evaluate our approach using the FrozenBiLM architecture [27] and compare it against the top-performing baselines based on two key metrics: average performance and the ratio of filtered data size to full data size (CC3M in this case). The average performance is calculated as the mean of the Recall at 1, 5, and 10 across four distinct downstream tasks.

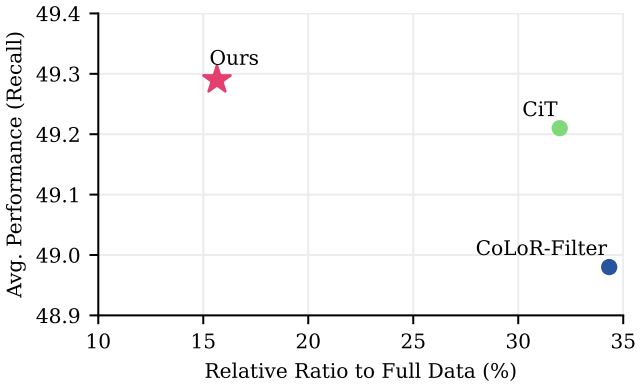


Figure 5. **Online multi-dataset (randomly shuffled VideoCC3M + WebVid2M) filtering and training performance comparison.** We compare our approach to the top performing baselines based on the average performance and the ratio of filtered data size to full data size (VideoCC3M + WebVid2M here). The average performance is the average of Recall at 1, 5 and 10 across five downstream tasks.

5. Extended Analysis

5.1. Embedding Robustness Analysis

Fig. 6 presents a comparative evaluation of various baseline methods using three distinct variants of CLIP embeddings: Max, Mid, and Avg. Across all embedding types, our proposed method consistently outperforms competing approaches, achieving the highest average recall while making use of small data subsets for training.

Relevance Filter	Clip ratio (%)	Avg. perf.
Cosine similarity (threshold = 0.55)	49.99%	48.55
Gaussian distribution modeling	45.51%	48.23
vMF distribution modeling	34.12%	48.95
vMF kernel density estimation (ours)	**27.50%**	**49.11**

Table 3. **Ablation of relevance filters on WebVid2M**

Threshold τ_{text}	Clip ratio (%)	Avg. perf.
$\tau_{\text{text}} = 0.5$	57.4%	48.58
$\tau_{\text{text}} = 0.55$	42.5%	**48.99**
$\tau_{\text{text}} = 0.6$	29.9%	48.89
$\tau_{\text{text}} = 0.65$	**20.1%**	48.62

Table 4. **Ablation on the choice of text cosine similarity threshold τ_{text} in CiT [26].** $\tau_{\text{text}} = 0.55$, which is the default value in the original CiT paper and the value we use for our main experiments, shows the best performance. The results are from WebVid2M.

These results show the robustness of our approach, demonstrating its effectiveness across diverse embedding variants, in addition to the primary LanguageBind embeddings used in our main experiments. This consistent performance further highlights the versatility of the proposed method, emphasizing its potential for broad applicability across a range of tasks and datasets.

5.2. Relevance Filter Ablations

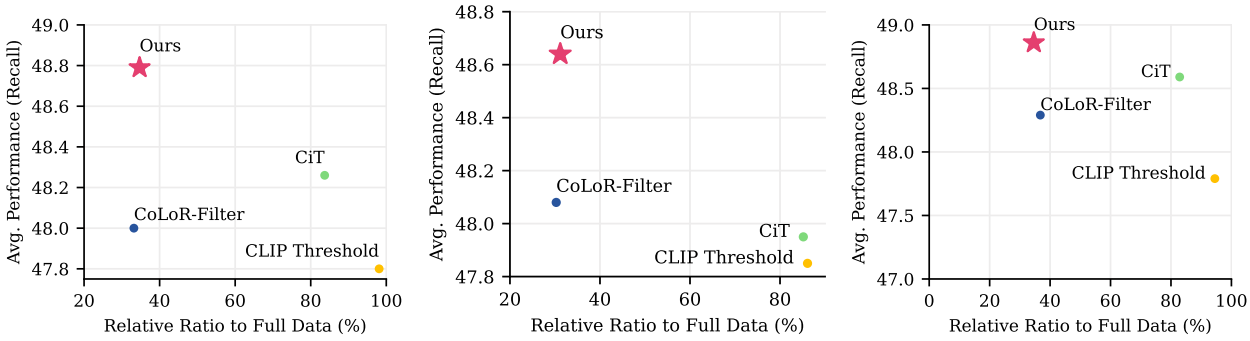
To better understand and justify the design choice of our relevance filter, we conduct three comparisons. The first comparison involves using cosine similarity on downstream data, the second compares modeling each downstream task using a Gaussian distribution, and the third explores the use of a von Mises-Fisher (vMF) distribution to model each downstream tasks. As shown in Table 3, our relevance filter with vMF kernel density estimation outperforms the other approaches by a significant margin while requiring the least amount of data.

5.3. Baseline Ablations

CiT [26] requires the hyperparameter τ_{text} that is used as a threshold on the cosine similarity between incoming data and downstream task text embeddings. Tab. 4 shows the ablation on the choice of τ_{text}. As in the main experimental results of the original CiT paper, choosing $\tau_{\text{text}} = 0.55$ shows the best performance.

For another ablation, we experiment with CiT without the parameter update, which we call train-free CiT, and the results can be found in Tab. 5. While the train-free CiT improves the computational efficiency over CiT, as it does not require any parameter update of the filtering model, its performance worsens in terms of both the average performance and the number of data sampled.

While we adapt CoLoR-Filter to operate in sample-wise manner to better suit the online video-text filtering setting,



(a) Using CLIP Max embedding (b) Using CLIP Mid embedding (c) Using CLIP Avg embedding

Figure 6. **Performance comparison using Max/Mid/Avg CLIP embeddings** We compare our approach to the baselines based on the average performance and the ratio of filtered data size to full data size. The average performance is the average of Recall at 1, 5, and 10 across the five downstream tasks. All experiments shown are based on WebVid2M.

Filtering method	Clip ratio (%)	Avg. perf.
CiT [26]	**42.5%**	**48.99**
Train-free CiT	51.5%	48.63

Table 5. **Ablation of CiT [26] with and without filtering model parameter update.** While the Train-free CiT, which does not require the parameter update of the filtering model, improves the computation efficiency, it results in worse performance while sampling more data. The results are from WebVid2M.

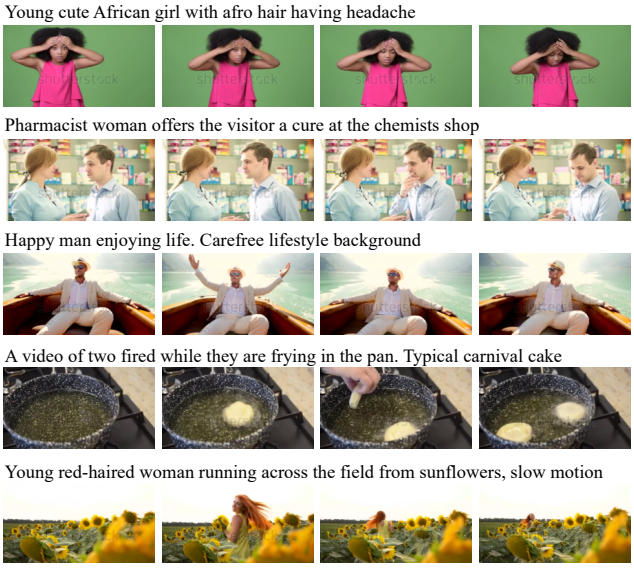
we also show the result of applying CoLoR-Filter in a batch-wise manner. In the batch-wise CoLoR-Filter, the incoming data is first stored in a delayed buffer (batch) if it passes the video-text cosine similarity thresholding. When the buffer is full, top $p\%$ of the data within the delayed buffer is selected based on the cosine similarity difference between the finetuned and the prior models. Tab. 6 shows the result of using batch-wise CoLoR-Filter in online video-text filtering with the buffer size of 100. Note that while the performance slightly improves, using the batch-wise CoLoR-Filter increases the storage cost and decreases the responsiveness as it needs to wait until the buffer is full, making it less efficient and applicable in the online filtering setting. It also introduces another hyperparameter p, the sampling ratio within the buffer.

5.4. Qualitative Analysis

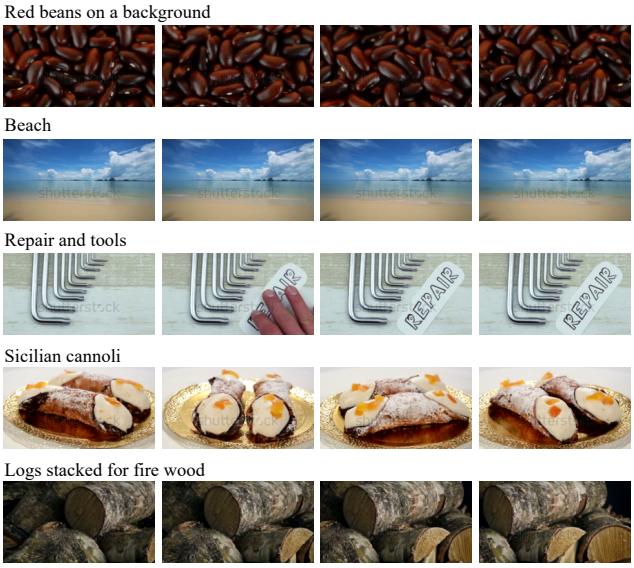
We present qualitative analyses in Fig. 7, Fig. 8, Fig. 9, Fig. 10 and Fig. 11. Fig. 7-(a) shows samples selected by ReSpec but not by any of the baselines. These samples are generally meaningful regarding alignment, downstream task relevance, and text specificity. However, the baseline fails to select these samples, indicating that the efficiency of task-aware online training is reduced. Fig. 7-(b) and (c) display samples selected by CiT and CoLoR-Filter but not by ReSpec. This suggests that while CiT and CoLoR-Filter maintain a certain level of downstream task relevance, they often select less informative samples with insufficient text specificity. Fig. 7-(d) demonstrates that the LB Threshold method selects data that maintains alignment but falls significantly short in relevance and text specificity.

Sampling ratio p	Clip ratio (%)	Avg. perf.
50%	38.7%	48.46
45%	34.8%	48.46
40%	31.0%	48.58
35%	27.1%	48.67
30%	23.2%	48.67
25%	19.4%	48.44
Sample-wise	56.4%	48.46

Table 6. **Results of using batch-wise version of CoLoR-Filter.** While using the batch-wise version slightly improves the performance, note that it incurs additional storage cost and reduces the responsiveness. The results are from WebVid2M.

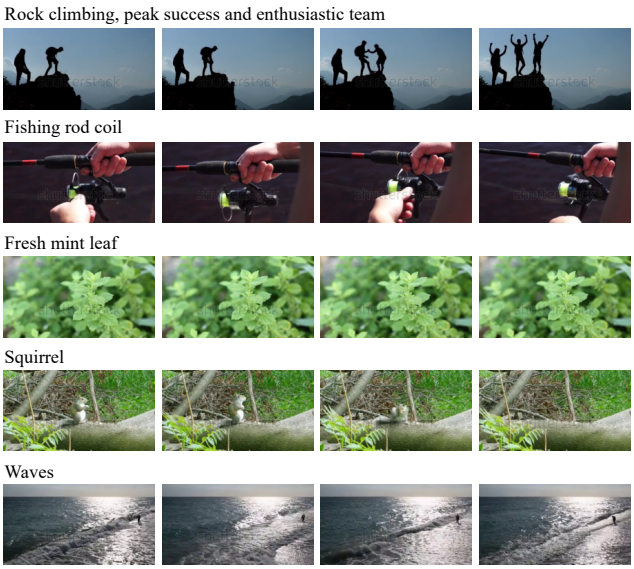


(a) Sampled by **ReSpec**, but not by all baselines



(b) Sampled by CiT, but not by **ReSpec**

(c) Sampled by CoLoR-Filter, but not by **ReSpec**



(d) Sampled by LB Threshold, but not by **ReSpec**

Figure 7. **Qualitative analysis** (a) represents samples only selected by ReSpec (ours) and not by other baselines (CiT, CoLoR-Filter). (b), (c), and (d) visualize samples selected by other baselines but not selected by ours. In each case, the samples selected generally ensure a certain level of alignment, relevance, and specificity. While downstream relevance is shown for (b) and (c), there are noticeable shortcomings in terms of text specificity. In the case of (d), there are shortcomings in both downstream relevance and text specificity. More samples are shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

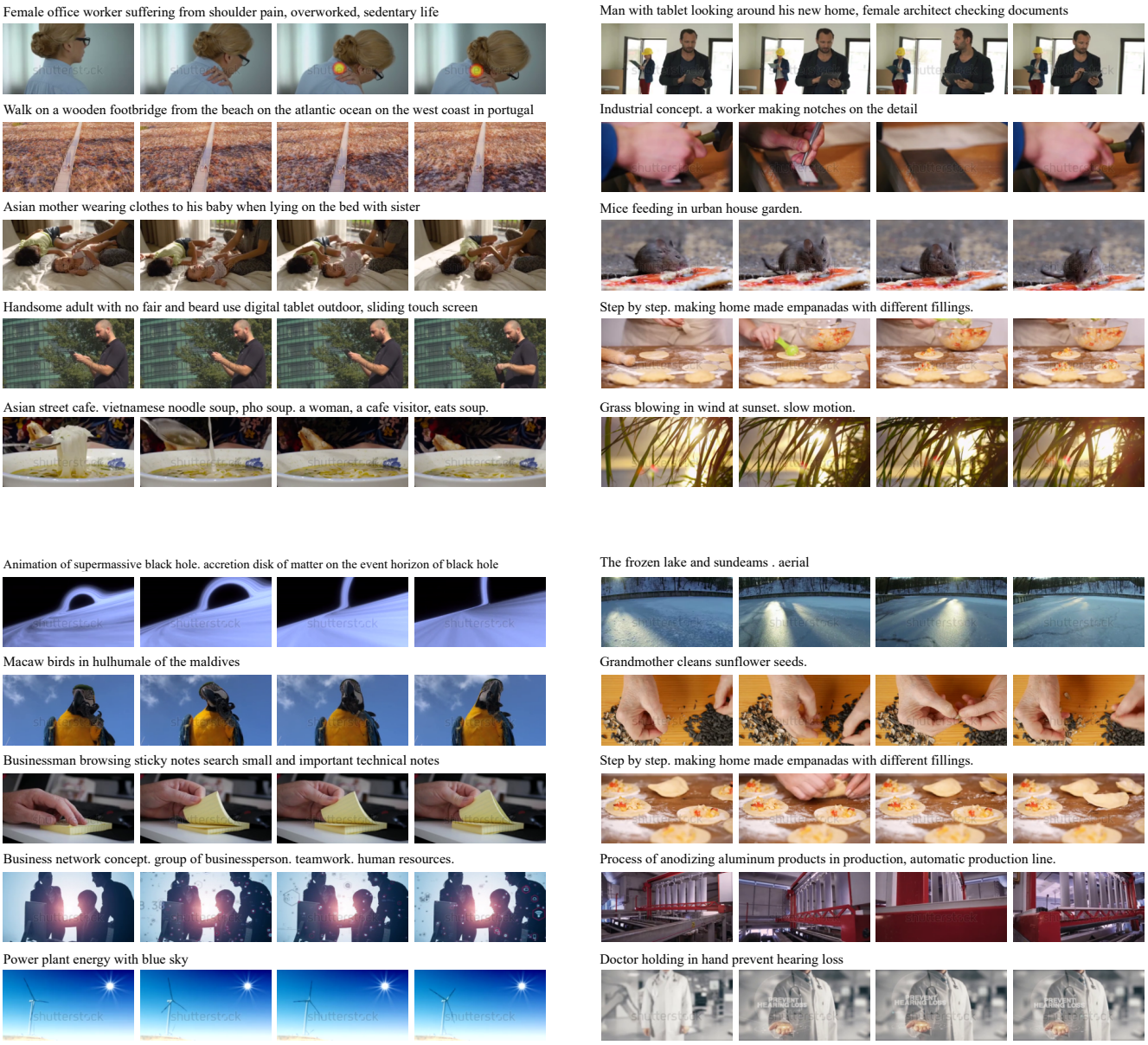


Figure 8. **Additional qualitative analysis of ReSpec**

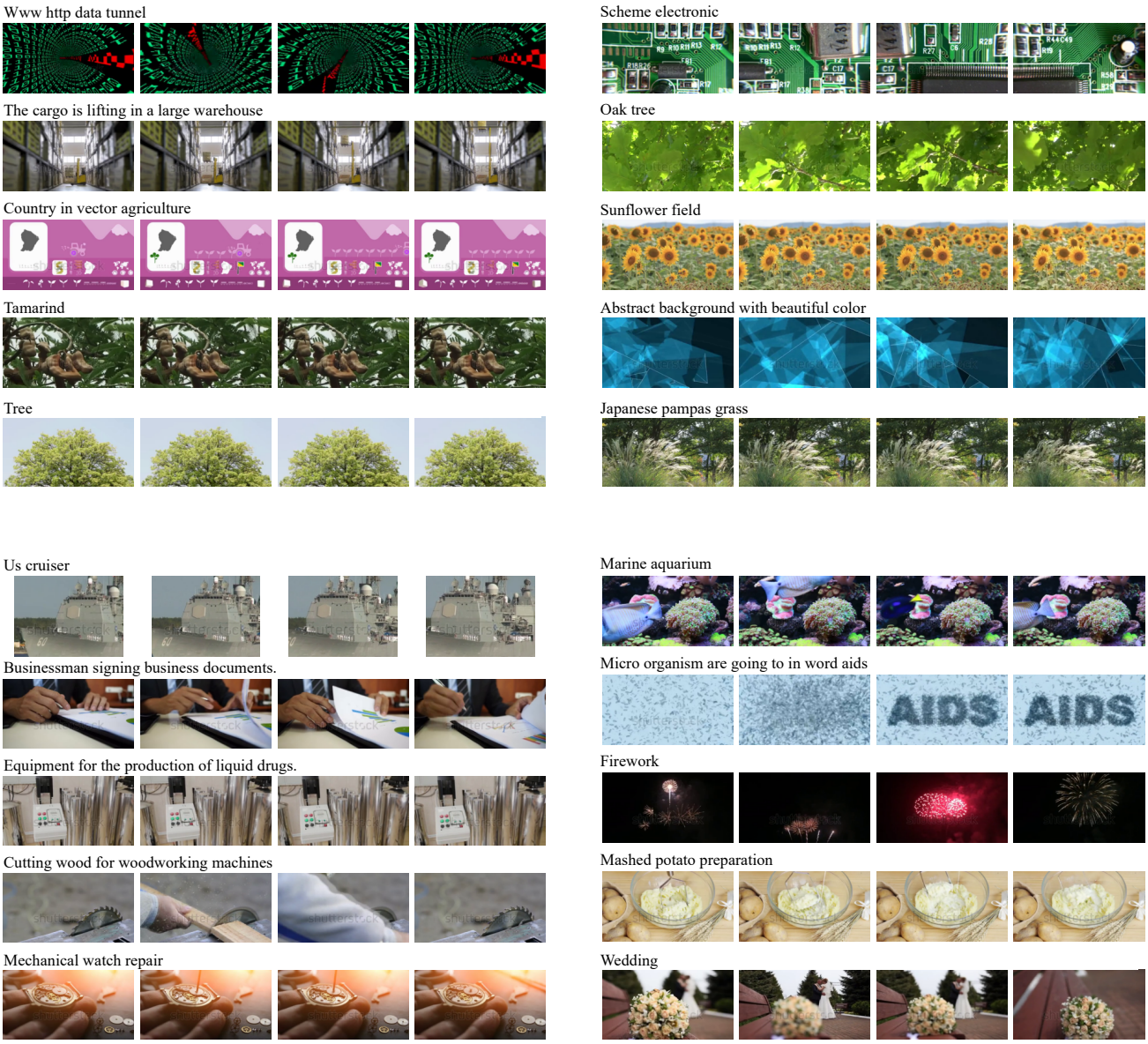


Figure 9. **Additional qualitative analysis of CiT**

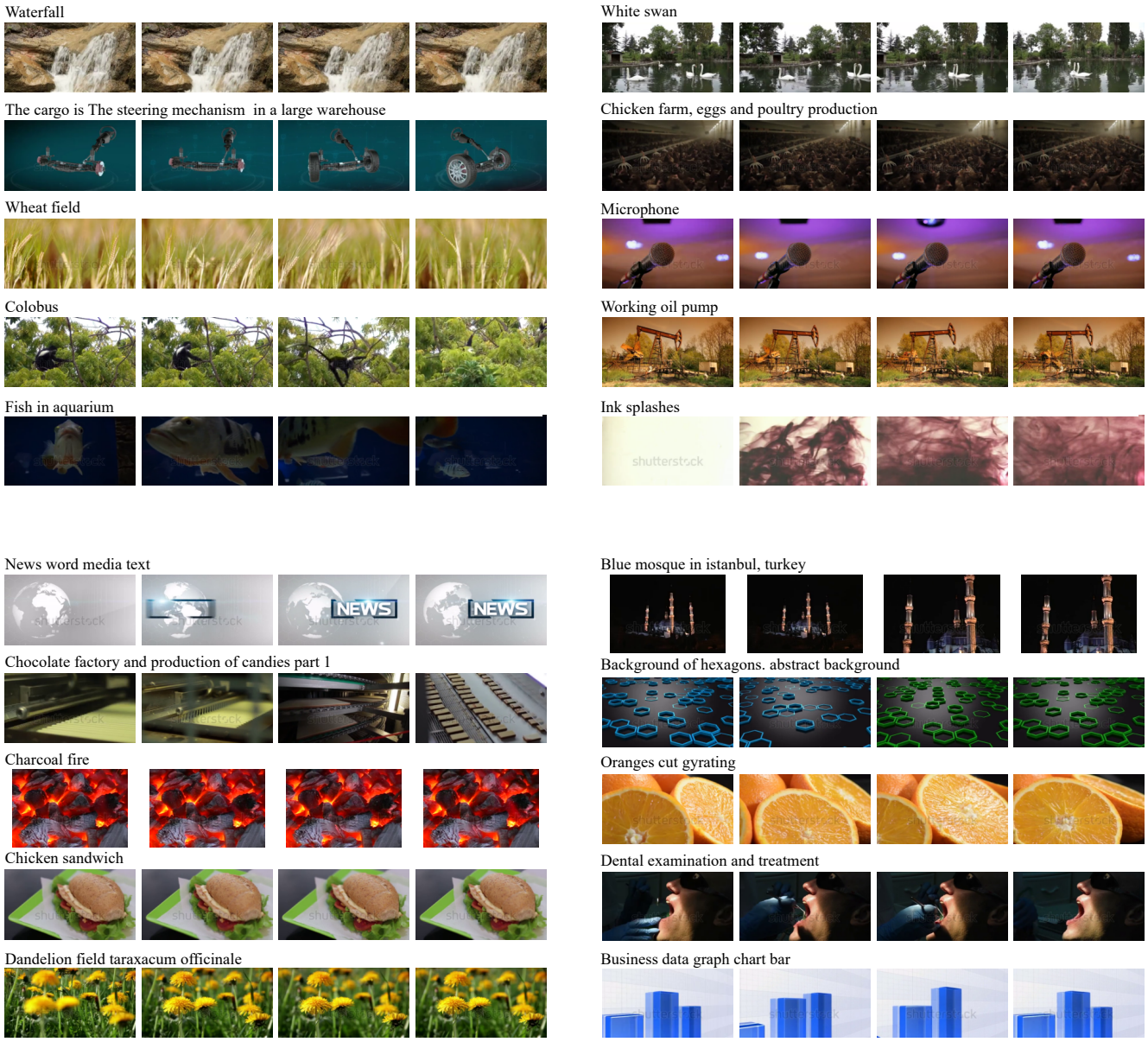


Figure 10. **Additional qualitative analysis of CoLoR-Filter**

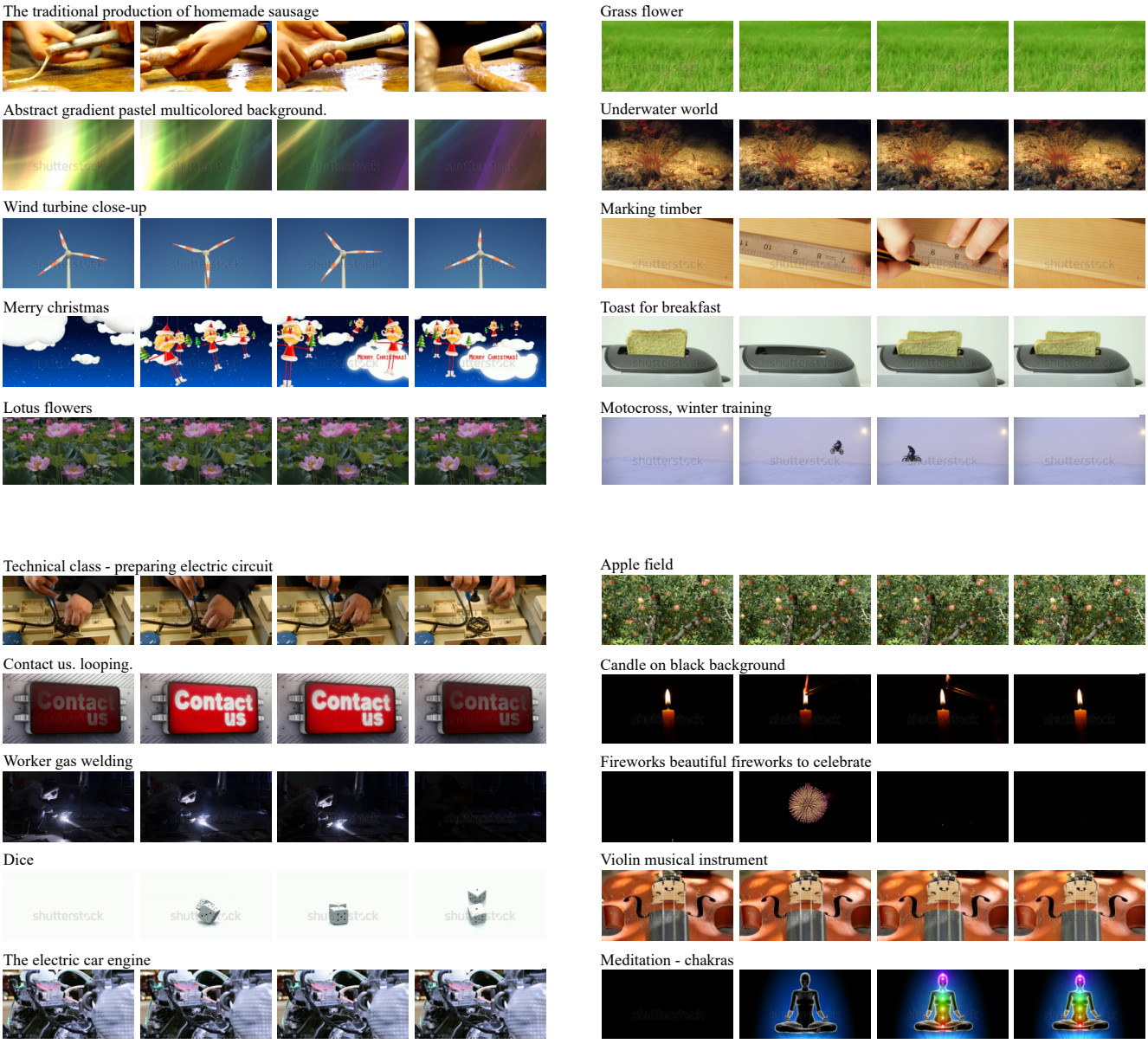


Figure 11. **Additional qualitative analysis of LB Threshold**

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *NeurIPS*, 2019. 1

[2] M. Bain, A. Nagrani, G. Varol, and A. Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 1

[3] Michele Boldo, Enrico Martini, Mirco De Marchi, Stefano Aldegheri, and Nicola Bombieri. On the query strategies for efficient online active distillation. *arXiv preprint arXiv:2309.01612*, 2023. 1

[4] David Brandfonbrener, Hanlin Zhang, Andreas Kirsch, Jonathan Richard Schwarz, and Sham Kakade. Color-filter: Conditional loss reduction filtering for targeted language model pre-training. *NeurIPS*, 2024. 2, 3

[5] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *ICCV*, 2021. 1

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 4

[7] Mohammad Farhadi, Mehdi Ghasemi, Sarma Vrudhula, and Yezhou Yang. Enabling incremental knowledge transfer for object detection at the edge. In *CVPR Workshops*, 2020. 1

[8] Zhenyu Huang, Guocheng Niu, Xiao Liu, Wenbiao Ding, Xinyan Xiao, Hua Wu, and Xi Peng. Learning with noisy correspondence for cross-modal matching. *NeurIPS*, 2021. 1

[9] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4

[11] Yijie Lin, Jie Zhang, Zhenyu Huang, Jia Liu, Zujie Wen, and Xi Peng. Multi-granularity correspondence learning from long-term noisy videos. *ICLR*, 2024. 1

[12] Ruyang Liu, Chen Li, Yixiao Ge, Thomas H Li, Ying Shan, and Ge Li. Bt-adapter: Video conversation is feasible without video instruction tuning. In *CVPR*, 2024. 2, 3

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[14] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*, 2017. 3

[15] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 2022. 1

[16] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 2

[17] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In *ICCV*, 2019. 1

[18] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *ECCV*, 2022. 1

[19] 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 *ICML*, 2021. 1

[20] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1, 4

[21] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022. 2

[22] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 1, 4

[23] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *CVPR*, 2024. 2

[24] Zhen Wang, Liu Liu, Yajing Kong, Jiaxian Guo, and Dacheng Tao. Online continual learning with contrastive vision transformer. In *ECCV*, 2022. 1

[25] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, 2017. 3

[26] Hu Xu, Saining Xie, Po-Yao Huang, Licheng Yu, Russell Howes, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Cit: Curation in training for effective vision-language data. In *ICCV*, 2023. 2, 3, 4, 5, 6

[27] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *NeurIPS*, 2022. 3, 5

[28] Yuncong Yang, Jiawei Ma, Shiyuan Huang, Long Chen, Xudong Lin, Guangxing Han, and Shih-Fu Chang. Tempclr: Temporal alignment representation with contrastive learning. *ICLR*, 2023. 1

[29] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 4

[30] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 3

[31] Xu Zhang, Hao Li, and Mang Ye. Negative pre-aware for noisy cross-modal matching. In *AAAI*, 2024. 1

[32] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *ICLR*, 2024. 2