Moment Quantization for Video Temporal Grounding

Supplementary Material

This supplementary material provides more details of our Moment-Quantization based Video Temporal Grounding method:

- Details about Training Objectives (Sec. A)
- Datasets and Training Details (Sec. B)
- Additional Experiments (Sec. C)

A. Details about Training Objectives

As mentioned in Sec.3.6, in addition to the moment quantization loss $\mathcal{L}_{\text{mq}}$, we also adopt the moment retrieval loss $\mathcal{L}_{\text{mr}}$, the highlight detection loss $\mathcal{L}_{\text{hd}}$, and the alignment loss $\mathcal{L}_{\text{align}}$ as supervision.

To predict the timestamp of the target moments, we utilize L1 loss with focal loss [4] to classify the moment queries between foreground and background. Given the ground truth moment $\hat{m} = (\hat{m}_c, \hat{m}_\sigma)$ and binary classification label $\hat{g}_{\text{c}}$, and corresponding predictions as $m = (m_c, m_\sigma)$ and p_{c}, respectively, where p_{c} is predicted by a two-layer 1D convolution module. Then the moment retrieval loss $\mathcal{L}_{\text{mr}}$ is formulated as:

$$\mathcal{L}_{\text{mr}} = \lambda_{\text{L1}} \,||\, \hat{m} - m \,||\, + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}(\hat{g}_{\text{c}}, p_{\text{c}}), \tag{1}$$

where λ_* are balancing parameters. $\mathcal{L}_{\text{cls}}$ is as follow:

$$\mathcal{L}_{\text{cls}}(\hat{g}_{\text{c}}, p_{\text{c}}) = \begin{cases} -\alpha(1-p_{\text{c}})^\gamma \log p_{\text{c}} & \text{if } \hat{g}_{\text{c}} = 1 \\ -(1-\alpha) p_{\text{c}}^\gamma \log\,(1-p_{\text{c}}) & \text{otherwise,} \end{cases} \tag{2}$$

where $\alpha = 0.9$ and $\gamma = 2$ are empirical hyperparameters. We use a single L1 loss instead of the combination of Smooth L1 Loss and GIoU Loss as noted in [3].

Following the previous method [3], we adopt intra-video contrastive loss as highlight detection loss $\mathcal{L}_{\text{hd}}$. To obtain the saliency scores $\hat{s}_i$ for HD, we calculate the cosine similarities between the pooled textual features $\tilde{q}$ and each token in the semantic-aware continuous video features z_t:

$$\hat{s}_i = \frac{z_t^\top \tilde{q}}{||z_t||_2 \, ||\tilde{q}||_2}. \tag{3}$$

Then we apply intra-video contrastive learning between sampled positive frames (with index $p \in P$) and the pooled textual query $\tilde{q}$:

$$\mathcal{L}_{\text{hd}} = -\log \frac{\exp(\hat{s}_p/\tau)}{\exp(\hat{s}_p/\tau) + \sum_{i\in\Omega} \exp(\hat{s}_i/\tau)}. \tag{4}$$

Here, Ω is the set of frame indices where $s_i < s_p$, and τ is a temperature parameter and set as 0.07.

Following [6], we apply a video-level constraint and a layer-wise constraint as the alignment loss. Because the lightweight recurrent structure from [6] leverages multi-layer CLIP features, we adopt InfoNCE loss to calculate the video-level and layer-wise constraints:

$$\mathcal{L}_{\text{video}} = \frac{1}{B} \sum_{b\in B} \text{InfoNCE}(\tilde{z}_t^b, \tilde{q}^b), \tag{5}$$

$$\mathcal{L}_{\text{layer}} = \frac{1}{N_l} \sum_{n\in N_l} \text{InfoNCE}(\tilde{z}_t^n, \tilde{q}^n), \tag{6}$$

where B denotes the batch size and N_l denotes the number of layers of CLIP feature. Here, $\mathcal{L}_{\text{video}}$ performs contrast among samples in the same batch and averages the loss across layers. $\mathcal{L}_{\text{layer}}$ performs contrast among layers and averages across the batch. Thus the alignment loss $\mathcal{L}_{\text{align}}$ is:

$$\mathcal{L}_{\text{align}} = \lambda_{\text{video}} \mathcal{L}_{\text{video}} + \lambda_{\text{layer}} \mathcal{L}_{\text{layer}}. \tag{7}$$

where λ_* are balancing parameters.

B. Datasets and Training Details

B.1. Datasets

Moment Retrieval. **QVHighlights** is a relatively recently publicized dataset by [2]. Consisting of varying lengths of moments and diverse text queries, it is a challenging and only dataset for joint moment retrieval and highlight detection tasks. It contains 10,148 videos and each is 150 seconds long. The training set, validation set and test set include 7,218, 1,550 and 1,542 video-text pairs, respectively. **Charades-STA** is annotated by [1] on Charades datasets using semi-automatic methods. In total, the video length is 30 seconds on average. There are 12,408 and 3,720 query-moment pairs in the training and testing sets, respectively. **TACoS** is collected by [9] and consists of 127 videos on cooking activities, which are around 5 minutes on average. We adopt the same split as [7], which involves 9,790 pairs for training and 4,436 pairs for testing. **Ego4D-NLQ** contains 1.3K videos with 8-20 minutes durations under daily egocentric scenarios. 15.2K queries in the form of questions are annotated with precise moments.

Highlight Detection. **TVSum** composes 50 videos of various genres, e.g., news, documentary, and vlog. Obtained via crowdsourcing, it has 20 saliency score annotations per video. We follow the settings in [5, 8]. **YouTube Highlights** is composed of 433 videos from 6 domains: dog, gymnastics, parkour, skating, skiing, and surfing. We follow [3] for the settings, as well as the usage of the domain name as the text query.

Dataset	Epoch	Bs	Lr	Lr drop	K	N_l	λ_{L1}	λ_{cls}	λ_{hd}	λ_{cmt}	λ_{video}	λ_{layer}
QVHighlights	60	64	$5e^{-4}$	20	1024	2	0.2	2.0	0.1	0.25	0.1	0.05
Charades-STA	50	16	$2.5e^{-4}$	30	1024	4	0.2	1.0	0.01	0.25	0.1	0.1
TACoS	100	8	$2.5e^{-4}$	50	1024	3	0.2	2.0	0.05	0.25	0.1	0.05
Ego4D-NLQ	60	11	$2.5e^{-4}$	20	1024	4	0.2	1.0	0.1	0.25	0.1	0.1
YouTube Highlights	200	4	$5e^{-4}$	–	Tab. 2	Tab. 2	–	1.0	0.1	0.25	0.1	0.1
TVSum	500	4	$5e^{-4}$	–	Tab. 3	Tab. 3	–	1.0	0.1	0.25	0.1	0.1

Table 1. Training details. We provide elaborate training details on each dataset. Bs denotes batch size; Lr denotes learning rate; Lr drop denotes the drop of learning rate at the specific epoch. K denotes the size of the moment codebook. N_l denotes the number of layers of CLIP features we used.

Domain	Dog	Gym.	Par.	Ska.	Ski.	Sur.
K	512	512	1024	512	512	512
N_l	3	3	3	3	3	2

Table 2. K and N_l for YouTube Highlights.

Domain	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS
K	1024	1024	1024	512	256	256	1024	512	512	512
N_l	3	2	3	4	4	2	4	3	4	4

Table 3. K and N_l for TVSum.

Dynamic	Projector	R1		mAP		
		@0.5	@0.7	@0.5	@0.75	Avg.
✓		67.48	50.97	68.33	49.98	47.86
	✓	**68.13**	51.48	**68.81**	50.27	47.82
✓	✓	67.94	**53.03**	68.54	**51.48**	**48.81**

Table 4. The impact of maintaining a dynamic codebook and incorporating a projector.

Method	Params (M)	Memory (GB)	R1		mAP
			@0.5	@0.7	Avg.
baseline	**2.71**	**9.83**	65.35	49.42	45.63
Ours	3.02	9.97	**67.94**	**53.03**	**48.81**

Table 5. The comparison of model size in the training phase. Params and Memory represent the number of learnable parameters and peak GPU memory (64 batch size), respectively.

Method	R1		mAP		
	@0.5	@0.7	@0.5	@0.75	Avg.
Modulated	67.81	52.39	68.07	50.29	48.02
Continuous	**67.94**	**53.03**	**68.54**	**51.48**	**48.81**

Table 6. The impact of maintaining a modulated codebook.

B.2. Training Details

Elaborate parameter settings for each benchmark are summarized in Tab. 1, Tab. 2 and Tab. 3. For YouTube HL and TVSum, we achieve the best performance through hyperparameter tuning. The same codebook size across different domains could achieve similar performance to the paper.

After moment quantization, a temporal feature pyramid [6] is constructed by applying 1D convolutions with on the semantic-aware video features z_t. The strides of the pyramid are set to (1, 2, 4, 8) by default. We concatenate features from all levels to predict once in all heads. Automatic mixed precision (AMP) with FP16 is utilized to accelerate training. For TVSum and YouTube Highlights, we adopt random initialization on the codebook since limited clip-level features cannot generate enough cluster centers.

C. Additional Experiments

C.1. Dynamic and Projected Codebook

As described in Sec.3.6, the moment codebook is initialized using a CLIP encoder to extract clip-level features on each dataset. During the training, we optimize a projector to map the entire codebook to a latent space. Tab. 4 illustrates the comparison with two alternatives: 1) omitting the projector; and 2) making each entry in the initialized codebook static. Our results show that incorporating the projector significantly improves performance.

C.2. Computational Consumption

We compare model size in terms of the number of learnable parameters and peak GPU memory. As shown in Tab. 5, our moment quantization method significantly improves performance with only a slight increase in model size.

C.3. Modulated Codebook

The results indicate that directly using discrete features is ineffective. Therefore, we explore another indirect method in Tab. 6. We follow the modulated quantization operation proposed in [10], first normalizing the continuous features and then modulating them with learned scales and biases computed from the discrete features. This modulated quantization is more effective than directly using discrete features but still lags behind our soft quantization.

References

[1] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017. 1

[2] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858, 2021. 1

[3] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804, 2023. 1

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1

[5] Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3042–3051, 2022. 1

[6] Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. R2-tuning: Efficient image-to-video transfer learning for video temporal grounding. In *European Conference on Computer Vision*, pages 421–438. Springer, 2024. 1, 2

[7] WonJun Moon, Sangeek Hyun, SuBeen Lee, and Jae-Pil Heo. Correlation-guided query-dependency calibration in video representation learning for temporal grounding. *arXiv preprint arXiv:2311.08835*, 2023. 1

[8] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023. 1

[9] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. 1

[10] Chuanxia Zheng, Tung-Long Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35:23412–23425, 2022. 2