Supplementary Material for Variational Stacked Local Attention Networks for Diverse Video Captioning

Tonmoay Deb Akib Sadmanee Kishor Kumar Bhaumik Amin Ahsan Ali
M Ashraful Amin A K M Mahbubur Rahman

Artificial Intelligence and Cybernetics (AGenCy) Lab, Independent University, Bangladesh
tonmoay.nsu@gmail.com, {1620274, 1621366, aminali, aminmdashraful, akmmrahman}@iub.edu.bd

Abstract

This supplementary material includes more dataset details and initial preprocessing steps. Also, this material further extends our analysis and provides the reasoning behind choosing a specific loss criterion, i.e., Shared Learning Loss instead of RL or XE. After that, we grounded our empirical selection of η to balance XE and RL loss in the Shared Learning Loss. Further studies include analysis of using other pre-trained models for feature extraction and impact on our results. We have performed two experiments to demonstrate our model's robustness, analyzing results on shuffling feature sets and evaluating VSLAN performance after applying a similar feature set multiple times. The final experiment was on cross-dataset, where we trained VSLAN on one dataset, e.g., MSVD, and tested on MSR-VTT (and vice-versa) to evaluate performance consistency.

1. Dataset Details and Preprocessing

We evaluate VSLAN on MS Research Video Description (MSVD/ YouTube2Text) [2] and MSR-VTT [7].

MSVD/YouTube2Text [2]: This dataset contains 1970 singular-activity, short YouTube open domain video clips in total. Each clip comprises an average of 9.6 seconds of videos and has around 40 multilingual, human-annotated captions. We use only the English language corpus, which has a 12,594 vocabulary size. For experimental setup, according to the prior works [6], we split the dataset as 1200 clips for training, 100 for validation, and 670 for the test.

MSR-VTT [7]: This is the largest video captioning dataset till date, with respect to number of videos, domain diversity, and the vocabulary size. The dataset contains 10,000 video clips out of 7,180 videos, categorized into 20 contexts. The average clip duration is 14.8 seconds, and each clip holds 20 single sentences, comprising a total of 200,000 sentences and 29,316 corpus vocabulary sizes. According to the recommended setup by [7], we use 6,513 clips for training, 497 for validation, and 2,990 for testing.

Data Preprocessing: Firstly, we convert the sentences of both datasets into lower-case, truncated punctuations, and tokenized them based on singular space. Next, We select 9,657 most frequent words from the vocabulary of MSVD and 23,500 for MSR-VTT. Later, we index the captions based on the refined vocabulary indices. Additionally, we append unknown token $< u >$ to the vocabulary to index the words not present within the range. We append start $< b >$, end $< e >$ tokens at the beginning and end of each caption. During training, maximum sentence length U is set according to the highest U for the entire batch. Shorter sentences are padded with $< p >$. For the inference stage, we set the maximum sentence length U to 25 for both datasets. We apply NLTK [1] tool on the processed sentences to extract POS data for training our POS encoder network, VaPEn.

2. Visual Analysis of Model Convergence

2.1. Impact of loss calculation methods

In Figure 1, we can see a comparison of loss values up to 50 epochs of each method we discussed in the main paper. We can see some interesting insights from the plot. First of all, the Cross-Entropy (XE) loss is very high initially, and it converges over time, but not better as the Reinforcement Learning (RL) loss. As mentioned earlier, XE loss emphasizes the sentence structure, whereas RL loss focuses on each sentence's meaning with entailment rewards. For this reason, the convergence rate of RL loss is faster than XE. However, we can notice an interesting phenomenon in Shared Learning (which combines both XE and RL). For this reason, from the initial step, the combined loss picks essential information from both losses, and the loss converges faster. Also, this loss is relatively stable compared to XE and RL. During the 24 epoch, the model achieved the

[1] https://www.nltk.org

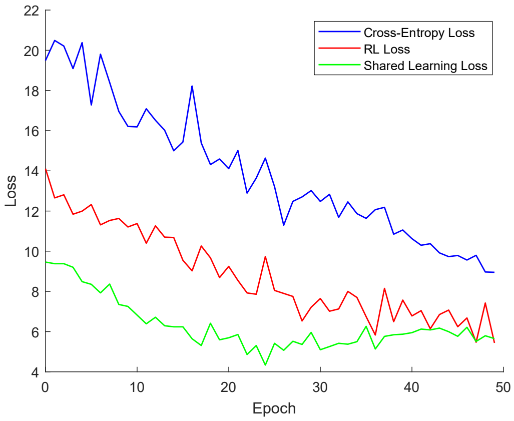

Figure 1. Comparison of the loss values (lower is better) upon each training epoch for each of the methods

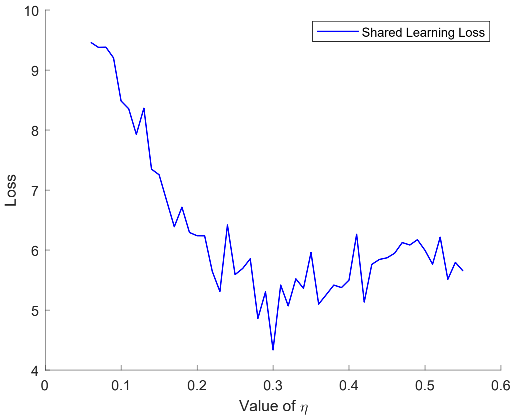

Figure 2. Comparison of the loss values (lower is better) upon each training epoch for each of the methods

lowest loss, and the Shared Learning Loss increased later. This phenomenon occurred because of the instability of the RL training [5], which is still an open problem.

2.2. Impact of η

In the earlier section, we have visualized the loss functions' performance and noted that Shared Learning loss converges best. However, as mentioned in the main paper, we empirically picked $\eta = 0.3$ for all further experiments. Our reason behind this empirical selection of η can be demonstrated by Figure 2. Here, we can see that when we select a lower value of η, the model weights more on the RL loss than the XE. As discussed in the earlier section regarding the instability of RL, the overall loss is also high. However, once we increase the value of η, the model becomes more stable, and for $\eta = 0.3$, we have the lowest loss. Although increasing η further increases the loss, it is not so high as RL because XE loss is relatively more stable than RL.

From these two analyses above, we can note that although RL loss helps is faster convergence, it alone can not guarantee the best performance. The combined approach of RL and XE with a proper η can be a wise selection.

3. Ablation Studies

We benchmark VSLAN by four additional experiments.

3.1. Impact of Sophisticated Pre-trained Models

We have evaluated VSLAN on four main feature sets, ResNext-101, C3D, VGG-16, and Faster-RCNN, to have identical feature sets with the compared methods. Here we will re-evaluate VSLAN on some more sophisticated 3D CNN visual features, SlowFast [3], and I3D [1]. ResNext-101 [4] exploits residual network with extended convolutional filter, where I3D inflates two-dimensional kernel into three and passes through the layers. For the experiments, we use the non-local version of the I3D pre-trained model on the Kinetics-400 dataset. SlowFast is a relatively newer approach, which splits a video clip into two streams, 'slow' and 'fast,' to capture the latent action properties. The 'slow' stream incorporates the spatial features, and the 'fast' stream represents the temporal features.

In Table 1, if we replace the 2D CNNs (VGG-16 and Faster R-CNN) with I3D and SlowFast, we can notice a significant performance drop. This is because VSLAN relies on object-related information (a core component of Faster R-CNN) alongside actions for generating a sentence. As no object-based 2D CNN models are not present in the second row of Table 1, the performance plummets. In this regard, if we replace I3D with Faster R-CNN, we notice a dramatic improvement in almost all matrices. Further, if we replace C3D with I3D, we can not notice any significant increment in the results. From here, we can note that VLAN distills information from 2D and 3D CNNs, and a fair balance between these may result in the best performance. Also, we claim that regardless of the visual features, the performance pattern was identical in both MSVD and MSR-VTT datasets. In further analysis, we set L^0= ResNext-101, L^1= Slowfast, L^2=I3D, and L^3= Faster R-CNN.

3.2. Shuffle Feature Set Order

In this experiment, we shuffle the order of features L^{0-3} to verify the robustness of the attention sub-networks. Other than the default order, $L^{(0,1,2,3)}$, we analyze the effects of scores on 4 distinct orders mentioned in Table 2. For $L^{(3,1,2,0)}$, where the model starts learning from R-CNN, drops CIDEr score by 3.5%. This phenomenon can be explained by the relative attention distribution of the model, which prioritizes object information and loses action syntax during caption generation. Because, $L^{(1,0,2,3)}$ yields better performance by 0.91% due to setting SlowFast at the initial stage. Second, depending widely on action representation, then leveraging object features can reduce the over-

	MSVD				MSR-VTT			
Model	**B@4**	**M**	**C**	**R**	**B@4**	**M**	**C**	**R**
RN, C3D, V, Faster R-CNN	57.4	36.9	98.1	75.6	46.5	32.8	55.8	62.4
RN, C3D, I3D, SlowfFast	54.9	35.1	89.8	73.1	42.2	30.8	53.9	59.8
RN, C3D, SlowFast, Faster R-CNN	57.8	37	**98.2**	75.7	46.5	33.1	56.5	62.9
RN, SlowFast, I3D, Faster R-CNN	**58**	**37.1**	98	**76**	**46.8**	**33.7**	**56.9**	**63**

Table 1. A relative comparison of the impact of pre-trained CNN models on VSLAN performance. RN is ResNext-101

all score. We note that $L^{(2,0,1,3)}$ propagates I3D followed by ResNext-101, which lowers the performance due to having two back-to-back object-rich information, SlowFast, and Faster R-CNN. Considering the drawback, in $L^{(0,3,2,1)}$, when we set SlowFast at the last stage, we notice a slight score gain by 0.18%. Due to holding rich spatial and temporal features, SlowFast fine-tunes both subject and action representation before sending features to the decoder.

3.3. Analysis on Similar Feature Set Stacking

To this extent, in the last 4 rows of Table 2, we analyze the performance of SlowFast by stacking up layers with redundant SlowFast feature information. We notice that, though adding similar features over the layers, the scores slightly increase up to 3 stacks. This output can be explained by higher-order feature interaction by bilinear pooling, which captures relevant information even with redundant data. However, due to model over-fitting, stack size 4 downgrades the overall scores, which is an open problem exploring the trade-off between dataset and model complexity. Interestingly, when multiple feature extractors were added, the performance climbed up. From here, we can infer that the FAN discounted the redundant visual features. In addition, the mBleu-4 column indicates the caption diversity of the models. We see that $L^{(0,3,2,1)}$ achieves the highest result due to an appropriate setting for the VaPEn.

Model	**B@4**	**M**	**C**	**R**	**mBleu-4**
Shuffled Order of Feature Sets (LAN decoder is present)					
VSLAN$_{\text{full}}$ - $L^{(3,1,2,0)}$	45.8	32.4	54.9	61.6	0.65
VSLAN$_{\text{full}}$ - $L^{(1,0,2,3)}$	46.1	**33.9**	55.4	62	0.67
VSLAN$_{\text{full}}$ - $L^{(2,0,1,3)}$	45.9	32.3	54.7	62.1	0.65
VSLAN$_{\text{full}}$ - $L^{(0,3,2,1)}$	**47**	33.7	**57**	**63.1**	**0.63**
Single Feature Set (SlowFast) with Identical Layer Stack					
VSLAN$_{L^1}$(SlowFast) ×1	40.5	27.7	45.1	58.8	0.71
VSLAN$_{L^1}$ (SlowFast) ×2	42.2	29.4	48.9	**59.2**	0.69
VSLAN$_{L^1}$ (SlowFast) ×3	**43.8**	**30.6**	**50.3**	57.5	0.69
VSLAN$_{L^1}$ (SlowFast) ×4	43.1	30.2	49.4	57.2	0.68

Table 2. Comparison of shuffled feature sets of VSLAN (first 4 rows) and stacked layer with identical SlowFast features

3.4. Cross-Dataset Analysis

With this analysis, our main goal is to make a statement that VSLAN trained model is consistent for 'out-of-dataset' information. For example, we expect to have a stable performance on the MSR-VTT test set even if we train VSLAN on the MSVD dataset (and vice-versa). If the performance on VSLAN is consistent, we can claim that the inherent model is learning to generate captions that will not be restricted to the trained dataset only. Table 3 visualizes the performance on cross-dataset. Interestingly, when we train on MSR-VTT and test on MSVD, we have a higher score gain than training on MSVD. This is because the MSVD dataset is smaller, and the number of captions per video in MSR-VTT is higher than MSVD. However, compared with the existing methods' outcomes, we can see that VSLAN outperforms POS-CG on MSR-VTT by 3.8% CIDEr and MSVD by 3.6%. This phenomenon can also be noted for the other two methods. Although compared models utilize multiple and identical features set by VSLAN, they fall short in the inherent architecture that VSLAN utilizes to combine those feature sets for robust performance.

4. Conclusion

We have performed a comprehensive evaluation of VSLAN from several aspects and grounded the stability of our method with the experimental outcome. Moreover, we have attached the code file for a sample reproduction of Table 2's row 5.

Model	**B@4**	**M**	**C**	**R**
The Output on MSVD Dataset Test Set with Training on MSR-VTT				
RecNet$_{\text{local}}$	53.5	35.4	85.9	71.6
GRU-EVE	50.1	36.5	81.8	72
POS-CG	56.5	36.8	95	74.7
VSLAN (ours)	**58.1**	**37.5**	**98.6**	**76.4**
The Output on MSR-VTT Dataset Test Set with Training on MSVD				
RecNet$_{\text{local}}$	35	24.3	37.5	50.2
GRU-EVE	33.8	25.1	41.8	51.7
POS-CG	36.1	25.8	44.3	55.1
VSLAN (ours)	**38.6**	**26.9**	**45.9**	**57.4**

Table 3. Cross-Dataset comparison with the closest competitors of VSLAN that use multiple feature streams

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, 2011.

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.

[4] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[5] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[6] Subhashini Venugopalan, Huijuan Xu, J. Donahue, Marcus Rohrbach, R. Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *HLT-NAACL*, 2015.

[7] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.