CM3T: Framework for Efficient Multimodal Learning for Inhomogenous Interaction Datasets - Supplementary Material

1. Training Details

For multi-head vision transformers, the bottleneck dimension used is $1/4$ multiplied by the channel dimension of the embedding for the respective block of the Video Swin-B. A smaller dimension size produces worse results, and a larger size produces similar results. For EK100, we use a slightly larger bottleneck dimension ($3/8$ times in place of $1/4$) for the last block of the Video Swin-B. For prefix tuning, the prefix channel dimension used is a minimum of 64 and $1/8$ multiplied by the channel dimension of the embedding for the respective block of the Video Swin-B. The bottleneck dimension for the generation of the prefixes is $1/4$ multiplied by the channel dimension of the prefix. Smaller prefixes provide worse results. Larger prefixes make the networks overfit very fast. The model starts overfitting after 5-6 epochs with a big prefix. Prefix tuning is a shortcut for the network to force attention to focus on particular features by learning fixed additional inputs to keys and values. This allows it to easily learn patterns in the inputs or activations of the training set and thus overfit.

For cross-attention, we use feature embeddings extracted from side modalities. We use trill-distilled [4] for obtaining audio features. Roughly 15 time steps of the audio features correspond to 128 frames in the videos (we use a stride of 4 frames for our input and Video Swin-B takes 32 frames as input). We use TV-L1 optical flow estimation and bninception [2] is used for its feature extraction. The features corresponding to each frame in the RGB video are taken, so the input of the temporal dimension is the same as RGB videos. Conv-1D is used for temporal pooling of all side modalities as Video Swin-B divides each input embedding into voxels and applies attention to each of them homogeneously. For simplicity, we give context from the side modalities for the whole input to every individual voxel. Also, we use performers [1] for cross-attention to make it more efficient.

The cross-attention adapters are added to the first two and last two layers of each block of Video Swin-B. For the last layer of the last block, we use traditional cross-attention by changing the input for the value to be the same as the key and use the cross-attention adapters for late fusion in place of modifying attention. This provides slightly better results as the information from the other modalities passes further along with the value.

For training, we use 8/16 Tesla V100 GPUs with a batch size of 3 per GPU for adapters and 2 for full finetuning. These are the largest batch sizes we can fit on one GPU for each case. We train for varying numbers of epochs, stopping if performance does not increase for 6 epochs. The learning rate is modified according to the batch size. Video Swin transformers use a batch size of 8 per GPU and a starting learning rate of 0.0003. CM3T uses 0.0015 for batch size of 2 and 0.0018 for batch size 3. The weight decay is 0.05. Weight initialization for downscaling weights is used as Kaiming initialization, zero initialization for upscaling. Rest weight initializations are either from the pretrained model or default initialization from PyTorch. We use Video Swin-B pretrained on SSv2 dataset for experiments on EK100 dataset. For the experiments on the other datasets, we use the same model pretrained on Kinetics400 dataset. SSv2 is an egocentric dataset, similar to EK100 and pretrained Video Swin-B uses a larger window size for it, so we chose it for EK100. Kinetics400 is similar to the other datasets, so we use it for experiments on the others.

2. Cross-attention adapters' behaviour with different modalities at different levels

We apply cross-attention to the first and last two layers of each block in video swin transformers [3] (each block has multiple stacked transformer encoders and each block has different spatial resolutions for the patches being processed). If the blocks have only two layers, we apply them to both the layers. In this subsection, we study the importance of cross-attention at different levels for different modalities, by removing adapters from different blocks. Table 1 shows the results. This can be used to prune the architecture for specific modalities. We see that for audio and transcript, later layers are more important, whereas for optical flow, earlier layers are more important. Block 3 is the biggest block and is needed for good results for all side modalities.

Table 1. Results for ablation study in section 2. The entries show cross attention removed from a particular block, this is represented by "-" in the table. For example, "CM3T - Block 1" represents CM3T without cross-attention in Block 1. Also, block 1 is closest to input and block 4 is the last block before classification head.

Method	Performance
Audio(MSE)	
UDIVA(CM3T)	0.69
UDIVA(CM3T - Block 1)	0.72
UDIVA(CM3T - Block 2)	0.73
UDIVA(CM3T - Block 3)	0.81
UDIVA(CM3T - Block 4)	0.78
Audio(Top-1 Accuracy)	
EK100(CM3T)	48.2%
EK100(CM3T - Block 1)	47.8%
EK100(CM3T - Block 2)	47.5%
EK100(CM3T - Block 3)	46.4%
EK100(CM3T - Block 4)	47.1%
Transcript(MSE)	
UDIVA(CM3T)	0.69
UDIVA(CM3T - Block 1)	0.70
UDIVA(CM3T - Block 2)	0.73
UDIVA(CM3T - Block 3)	0.82
UDIVA(CM3T - Block 4)	0.79
Optical Flow(Top-1 Accuracy)	
EK100(CM3T)	48.2%
EK100(CM3T - Block 1)	45.3%
EK100(CM3T - Block 2)	45.8%
EK100(CM3T - Block 3)	44.2%
EK100(CM3T - Block 4)	46.6%

Table 2. Result for adding adapters to cross-attention adapters.

Method	Performance
UDIVA(MSE)	
CM3T	0.690
CM3T (with adapters added to CA)	0.689

3. Adding adapters to cross-attention adapters

Since modalities are repeated across tasks and datasets, we see that training the entire cross-attention adapter module is not necessary. We can simply add scalable parallel adapters to the cross-attention modules. For this, the initial embedding is directly taken from the pretrained model and not the multi-head vision adapters, rest stays the same. Table 2 shows the results for this experiment. We train cross-attention adapters for audio using EK100 and show results for UDIVA with normal cross-attention adapters and adapters added to cross-attention adapters.

Table 3. Ablation study results. a_0 represents trainable scaling factor. a_1 represents changing activation function to ReLU.

Method	Performance
EK-100 (Accuracy(%))	
Scaled parallel adapters + PT	28.7
MHVA + PT	39.8
MHVA - a_0 + PT	38.7
MHVA - a_1 + PT	39.1

4. Ablation study: Our additions compared to adapters in Adaptformer

Table 3 shows the results of this ablation study.

References

[1] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. *CoRR*, abs/2009.14794, 2020. 1

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 1

[3] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211, June 2022. 1

[4] Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Félix de Chaumont Quitry, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, and Yinnon Haviv. Towards learning a universal non-semantic representation of speech. *Interspeech 2020*, Oct 2020. 1