Supplementary Material

Xiaotian Li Xiang Zhang Taoyue Wang Lijun Yin
State University of New York at Binghamton
{xli210, zxiang4, twang61, lyin}@Binghamton.edu

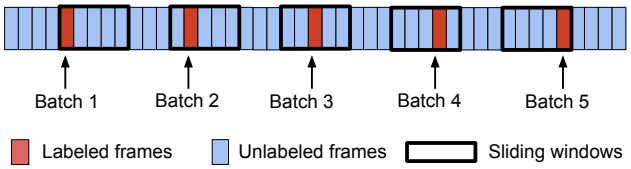

Figure 1: Illustration of sparsely labeled video clips. It shows how the location of the key frame shifts from 1 to n. Here $n = 5$.

1. Model

1.1. Details of the transformer-based relation encoder

The Spatial Teacher (i.e. Temporal Student) S_a, in Figure 2, is designed for modeling the spatial relationships between AUs. It is consisted of multiple identical blocks, and each block contains a multi-head self-attention module and a fully connected feed-forward layer. The original ViT uses an extra learnable BERT's token to perform the classification token, as their input images are split into fixed-size patch by location. In this work, the local region of AU-specific features have been fully activated before feeding them into the transformers. Thus, we remove the classification token design. $D_s = \left\{f_s^1, f_s^2, \cdots, f_s^u\right\} \in \mathbb{R}^{1\times U\times W}$ (Fig.2 in the original paper) where U indicates the AU number of the dataset, and W is the size of feature f_s. Here we set $W = U$. Likewise, $D_b = \left\{p_b^1, p_b^2, \cdots, p_b^n\right\} \in \mathbb{R}^{1\times N\times W}$ where N indicates the frame length of input clips, and W is the size of feature p_b. Here we set $W = U$. The dimension of positional embedding is consistent with f_s and p_b, and the dimension of initial frame-specific features p_a are consistent with the post frame-specific features p_b. The multi-head attention matrix (in the original paper) of the Spatial Teacher is denoted as $Head_i^s(Q_i, K_i, V_i) \in \mathbb{R}^{H\times U\times U}$ where U indicates the AU number of the dataset. H indicates the head number of the self-attention module. In this paper, we set 8 as the head number. The multi-head attention matrix of the Temporal Teacher is denoted as $Head_i^t(Q_i, K_i, V_i) \in \mathbb{R}^{H\times N\times N}$ where N indicates the frame length of input clips. H indicates the head number of the self-attention module.

1.2. Details of input data settings

KS sparsely samples the annotations by every k frames in the training data pool. Here we assume that only $\frac{1}{k}$ labels are available. KS sets the labeled frames as the key frames. These sparse key frames are fed into branch A for learning AU dependencies under fully supervision. At the same time, KS pick $n-1$ neighbours around the key frames as the unlabeled data. These neighbours, with the corresponding key frame, form an n-frame sequence as the input of branch B. As shown in Figure 1, we assume only one label is accessible by every $k = 9$ frames. The labeled frames (red color) denotes the key frames. For each video clip, it contains $n = 5$ frames. The key frame location is decided by $m = B \bmod n$ where B indicates the Bth batch of input data. Here the key frame location of the first batch of data is 1 mod 5 equals 1. As the batch number changing, the location of the key frame changes from 1 to 5 accordingly. Thus, each input video clip contains one label, and the data with different key frame location is balanced. The design is also used to decide the active student for spatial knowledge distillation.

1.3. Q&A for Model Design and Experiments

In this section, we explain some questions of the model design and experiments.

What's the difference between general Spatial-Temporal information and our Spatial-Temporal AU correlation knowledge? The multi-head attention matrices, in Figure 2, contains the relevance score of each atomic AU class and frame-level class. By modeling the spatial and temporal AU dependency in multiple attention matrices, the video-level and frame-level AU co-occurrence and mutual exclusive relation is refined and learned to improve the the general Spatial-Temporal information.

Why do we choose transformer? First, self-attention based methods (i.e., JAAnet [4]) have been proved to be very effective in learning AUs semantic relation. Second, the residual connection, multi-head attention, and positional

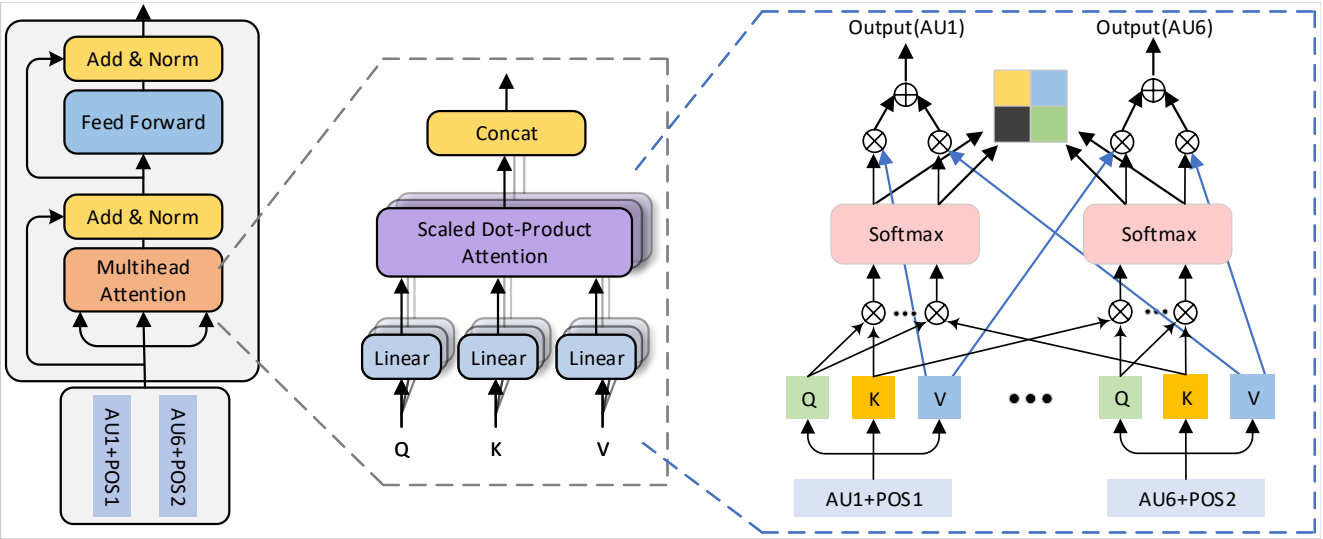

Figure 2: Illustration of the transformer-based model S_a for learning AU dependencies with the multi-head self-attention matrix. $\bigoplus$ indicates matrix addition. $\bigotimes$ indicates matrix multiplication.

embedding designs make it an efficient tool to learn the AU relation with multiple matrices (i.e. multiple complete graphs) and short or long-term temporal cues.

Why are Spatial Student models based on MLP instead of transformer? In most cases, KS need feed Spatial Student models with unlabeled data, which means the results derived by Spatial Students is less faithful. Thus, a weak design of Spatial Student is necessary. In additional, different design of Spatial Student amd Spatial Teacher can be recognized as a noise [5] or a model-wise perturbation for better applying the consistency regularization.

Why do we need the temporal constrain for pseudo-labeling? First, automatic AU detection, as a multi-label task, faces some difficulties in selecting and retaining the pseudo labels when only partial labels are with high confidence score. Compared with multi-class tasks, the one-hot format of multi-labels makes it hard to decide if the pseudo labels are confident in a holistic way or a local way. Setting the temporal constrain makes it easier to filter the cases which are not consist with the regular pattern of facial action movements. Second, the temporal label smoothness is also a soft constrains from human knowledge for better generalizing out-of-distribution AU data.

How to generate pseudo labels? We pick up the class which has maximum predicted probability for each binary class of unlabeled samples.

Can KS be trained with video clips without any labels? If yes, how? Yes, it can. When feeding KS with the unlabeled video clips, we only update our model with the loss of pseudo-labeling and Temporal Teacher. It worth noting that the unlabeled video clips are not allowed to use before the model training is stable (10 epochs in this project). Otherwise learning unreliable knowledge first will lead the model to the error-prone issue.

Why using 15% labels achieves nearly the same performance as using 100% on BP4D? This is due to the large portion of overlapped annotations, using 15% labels with sparsely sampled annotations makes the performance reach the "saturation" quickly, hence there is no significant performance gain after 15% towards the use of 100% labels. Our finding complies with the "less is better" principle confirmed by the other existing works [2].

How about applying different sequential perturbation for the pseudo label confirmation module? The fact is that using or mixing certain perturbations (e.g., temporal feature shift, random mask, and flip) does not bring obvious performance improvement. We speculate that other perturbations can not model the temporal fluctuations caused by incorrect pseudo labels well. Applying inappropriate or excessive perturbation operation can even degrade the performance.

1.4. Inference Strategy

We adopt the inductive learning for the proposed semi-supervised model. Only unlabeled data in the training dataset is used for pseudo-labeling. In the inference time, all the sub-networks are employed in the framework. The pseudo-labeling and the binary classification of perturbation-ware pseudo-labeling is removed from the framework.

With respect to the data structure, We adopt *sparse-training*-then-*dense-testing* [2] strategy for the proposed

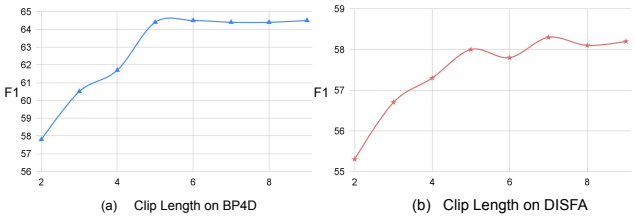

Figure 3: Effect of the video clip length n. (a) and (b) indicates the F1 score with different video clip lengths on BP4D and DISFA.

model. KS sparsely samples the annotations by every k frames in the training data pool. Then, the video clips are densely selected from the testing pool for predictions. The same key-frame position shifting strategy is applied in both training and inference stage.

2. Experiment

2.1. Additional Implementation Details

All training images in the same video clips are randomly rotated (-45 to 45 degrees), flipped horizontally (50% possibility), and with color jitters (saturation, contrast, and brightness) simultaneously. The detailed specification of Knowledge-Spreader is shown in the original code (model designing part). The complete code will be released to the research community by the time of the paper being published. We choose 5 as the frame length of each input video clip for the optimal time-and-accuracy trading-off. We implement our Knowledge-Spreader (KS) with the Pytorch framework and perform training and testing on the NVIDIA GeForce 2080Ti GPU.

2.2. Additional Quantitative Evaluation

The quantitative results with different label ratios are shown in Table 1 for reference. It corresponds to Figure 5 in the original paper. In addition, due to the page limitation, only partial comparison results with supervised methods in terms of individual AU are shown in Table 2 of the original paper. Table 2 shows the complete comparison results.

2.3. Effect of the Video Clip Length n

To investigate the influence of the input clip length, we perform experiments by the proposed model with 10% sparsely sampled annotations on BP4D and DISFA. Figure 3 shows the F1 score curve with n changes. Overall, the performance improve with the n increases from 2 to a certain threshold. A long video clip, on the other hand, results in high computational and memory costs. For optimal trading-off, 5 to 7 is a proper setting for the video clip length n.

Table 1: Quantitative comparison with semi-supervised methods using F1 score. Underlines indicate the best results of individual models.

Model	BP4D	DISFA	MME
Pseudo-label (1%)	54.3	40.4	45.8
Pseudo-label (2%)	57.8	50.8	47.5
Pseudo-label (5%)	59.7	51.5	52.1
Pseudo-label (10%)	60.7	56.8	54.2
Pseudo-label (15%)	61.2	57.1	54.9
Pseudo-label (20%)	62	58.5	55.2
Pseudo-label (50%)	63.6	57.9	55.3
Pseudo-label (60%)	62.7	56.7	55.3
Pseudo-label (70%)	63.3	57.9	55.3
Pseudo-label (80%)	62.4	58.3	56.6
Pseudo-label (90%)	62.3	57.5	55.5
Pseudo-label (100%)	62.7	58.8	56.9
Model	BP4D	DISFA	MME
FixMatch (1%)	49.9	35.6	41.6
FixMatch (2%)	55.1	46.2	46.5
FixMatch (5%)	59.2	52.7	52.6
FixMatch (10%)	60.5	55	55.4
FixMatch (15%)	62.1	57.7	55.6
FixMatch (20%)	62	58.4	56.4
FixMatch (50%)	62	57.9	58.3
FixMatch (60%)	62.1	56	56.4
FixMatch (70%)	61.9	57.8	57.2
FixMatch (80%)	62.2	56.9	55.5
FixMatch (90%)	61.9	57.5	55.3
FixMatch (100%)	62.7	58.8	56.9
Model	BP4D	DISFA	MME
TCL (1%)	55.6	42.3	43.3
TCL (2%)	58.9	51.2	48.2
TCL (5%)	60.5	53.6	53.4
TCL (10%)	61.7	55.8	55.7
TCL (15%)	62.3	56.7	56.2
TCL (20%)	62.7	57.9	55.6
TCL (50%)	63.2	59.2	57.6
TCL (60%)	62.8	60.1	57.9
TCL (70%)	63.0	59.6	57.9
TCL (80%)	62.9	60.4	58.3
TCL (90%)	62.7	58.3	57.8
TCL (100%)	63.1	59.7	58.1
Model	BP4D	DISFA	MME
Our KS (1%)	59.9	49.4	51.2
Our KS (2%)	62.5	52.8	54.8
Our KS (5%)	63.9	56.9	57.6
Our KS (10%)	64.4	58	58.4
Our KS (15%)	64.5	58.8	58.7
Our KS (20%)	64.4	59.5	58.9
Our KS (50%)	64.5	61.6	59.5
Our KS (60%)	64.4	62.9	59.4
Our KS (70%)	64.5	61.9	59.5
Our KS (80%)	64.4	62	59.4
Our KS (90%)	64.6	62.2	59.6
Our KS (100%)	64.7	62.8	59.7

2.4. Effect of the Perturbation-aware Pseudo-labeling

The module is consisted of two parts including Pseudo-labeling and a self-supervised module with loss function L_{self}. By removing the whole module, we observe the performance degradation by a margin of 0.8% and 1.0% on BP4D and DISFA. By only removing Pseudo-labeling,

Table 2: Comparison with state-of-the-art methods using F1 score in terms of individual AUs. The upper part is the F1 score on BP4D; The bottom part is the F1 score on DISFA. Bold numbers indicate the best performance.

Model	Used labels	AU1	AU2	AU4	AU6	AU7	AU10	AU12	AU14	AU15	AU17	AU23	AU24	Avg.
DSIN	100%	51.7	40.4	56.0	76.1	73.5	79.9	85.4	62.7	37.3	62.9	38.8	41.6	58.9
JAA	100%	47.2	44.0	54.9	77.5	74.6	84.0	86.9	61.9	43.6	60.3	42.7	41.9	60.0
LP	100%	43.4	38.0	54.2	77.1	76.7	83.8	87.2	63.3	45.3	60.5	48.1	54.2	61.0
ARL	100%	45.8	39.8	55.1	75.7	77.2	82.3	86.6	58.8	47.6	62.1	47.4	55.4	55.4
SRERL	100%	46.9	45.3	55.6	77.1	78.4	83.5	**87.6**	63.9	52.2	**63.9**	47.1	53.3	62.9
UGN	100%	54.2	46.4	56.8	76.2	76.7	82.4	86.1	64.7	51.2	63.1	48.5	53.6	63.3
HMP-PS	100%	53.1	46.1	56.0	76.5	76.9	82.1	86.4	64.8	51.5	63.0	49.9	54.5	63.4
FAUDT	100%	51.7	49.3	61.0	77.8	79.5	82.9	86.3	**67.6**	51.9	63.0	43.7	56.3	64.2
Our KS	15%	**58.7**	**50.3**	**62.0**	**79.5**	75.4	**84.9**	87.1	65.9	45.5	62.9	48.3	53.3	64.5
Our KS	100%	55.3	48.6	57.1	77.5	**81.8**	83.3	86.4	62.6	**52.3**	61.3	**51.6**	**58.3**	**64.7**

Model	Used labels	AU1	AU2	AU4	AU6	AU9	AU12	AU25	AU26	Avg.
DSIN	100%	42.4	39.0	68.4	28.6	46.8	70.8	90.4	42.2	53.6
JAA	100%	43.7	46.2	56.0	41.4	44.7	69.6	88.3	58.4	56.0
LP	100%	29.9	24.7	72.7	46.8	49.6	72.9	93.8	65.0	56.9
ARL	100%	43.9	42.1	63.6	41.8	40.0	**76.2**	**95.2**	66.8	58.7
SRERL	100%	45.7	47.8	59.6	47.1	45.6	73.5	84.3	43.6	55.9
UGN	100%	43.3	48.1	63.4	49.5	48.2	72.9	90.8	59.0	60.0
HMP-PS	100%	38.0	45.9	65.2	50.9	**50.8**	76.0	93.3	**67.6**	61.0
FAUDT	100%	46.1	48.6	**72.8**	**56.7**	50.0	72.1	90.8	55.4	61.5
Our KS	15%	41.7	53.5	69.7	41.3	46.2	72.0	92.3	54.0	58.8
Our KS	100%	**53.8**	**59.9**	69.2	54.2	**50.8**	75.8	92.2	46.8	**62.8**

the F1 score decreases by 0.6% and 0.7%. By replacing the self-supervised module with the hard threshold as the standard of confirming high-confident pseudo labels, we observe a performance drop by a margin of 0.3% and 0.4%. In addition, we compare the accuracy of pseudo labels generated by PPL and naïve pseudo-labeling [1] on BP4D using 10% labels. The result shows 76.35% accuracy on PPL and 73.36% on naïve pseudo-labeling. That demonstrates the performance of PPL improves by filtering the low quality pseudo labels with temporal perturbation. Another interesting finding is that if we only keep the loss L_{self} of PPL (not for label selection), the experimental results are also be reduced. That shows the auxiliary task in PPL benefit KS to learn better feature representation and inter-frame relation by identifying temporal disturbances.

2.5. Parameter Size Analysis

The trainable parameter size of the proposed model is around 25 million, which makes KS a very light-weighted model. Compared with the baseline algorithm EACnet [3], which contains 138 million parameters, Knowledge-Spreader, as a video-level model, reduces considerable parameter (80%) but achieves excellent performance improvement.

3. Dataset

3.1. Participants

233 participants were recruited from our University. There are 132 females and 101 males, with ages ranging from 18 to 70 years old. Ethnic/Racial Ancestries include Asian, Black, Hispanic/Latino, White, and others (e.g., Native American).

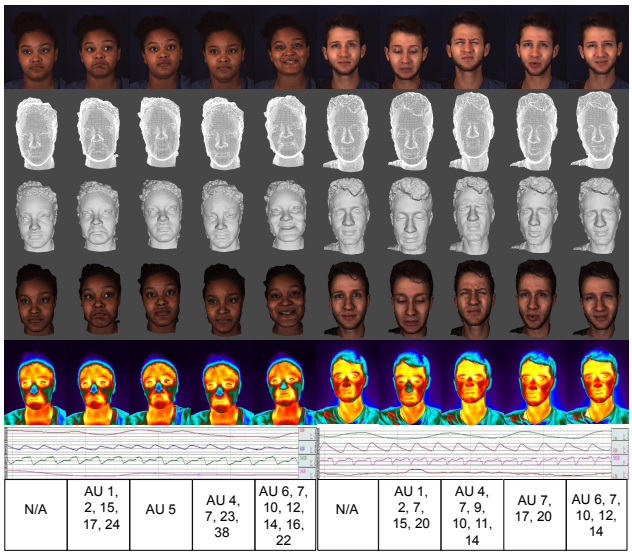

Figure 4: A sample sequence from our MME. 2D texture image, 3D mesh model, 3D shaded model, 3D texture model, thermal image, and physiological signal (respiration rate, blood pressure, EDA, heart rate) and corresponding AU occurrence are shown from top to bottom.

3.2. Recording System and Synchronization

Our data collection system consists of a 3D dynamic imaging camera system, a thermal sensor, a physiological signal sensor system, and a studio-quality audio recorder. The system setup and synchronization method are basically consistent with BP4D+ [6].

Table 3: The stimulus tasks designed for the data collection.

Task ID	Activity	Target Emotion
1	Have a pleasant chat with the interviewer	Happiness
2	Watch a 3D face model of the participant	Surprise
3	Watch an audio recording of 911 emergency call	Sadness
4	Experience a sudden sound from a horn	Startle or Surprise
5	React to a fake news	Skeptical
6	Asked to sing an impromptu song	Embarrassment
7	Experience physical fear of the threat in a dart game	Fear or Nervous
8	Experience the cold feeling by submerging hands into a bucket with ice water	Pain
9	React to the blame from the interviewer	Offended or Unpleasant
10	Experience a bad smell from decaying food	Disgust

3.3. Emotion Stimulus

Ten tasks were performed to elicit a wide range of spontaneous emotion expression (from positive, to neutral, and to negative) and inter-personal facial action behavior by a professional interviewer. Table 3 illustrates the detailed description for the designed tasks.

3.4. Data Organization

Each subject is associated with 10 different emotions and multi-modal data including the 3D sequence, 2D RGB sequence, thermal sequence, and the sequences of physiological data (i.e., blood pressure, EDA, heart rate, and respiration rate). The sample sequences of different modalities from two subjects are shown in Figure 4. Besides, the metadata including manually labeled action units occurrence and intensity, 3D/2D/IR facial landmarks, and 3D head poses are also generated for better analysis of automatic human facial action.

References

[1] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 2013. 4

[2] Jie Lei et al. 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 (CVPR)*, 2021. 2

[3] Wei Li et al. Eac-net: Deep nets with enhancing and cropping for facial action unit detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 4

[4] Zhiwen Shao et al. Deep adaptive attention for joint facial action unit detection and face alignment. In *Proceedings of ECCV*, 2018. 1

[5] Qizhe Xie et al. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 2

[6] Zheng Zhang et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4