arXiv:2312.07507v1 [cs.CV] 12 Dec 2023

NAC-TCN: Temporal Convolutional Networks with Causal Dilated Neighborhood Attention for Emotion Understanding

Alexander Mehta
alexandermehta@outlook.com
Independent Researcher
USA

William Yang
yangwill@seas.upenn.edu
University of Pennsylvania
USA

ABSTRACT

In the task of emotion recognition from videos, a key improvement has been to focus on emotions over time rather than a single frame. There are many architectures to address this task such as GRUs, LSTMs, Self-Attention, Transformers, and Temporal Convolutional Networks (TCNs). However, these methods suffer from high memory usage, large amounts of operations, or poor gradients. We propose a method known as **N**eighborhood **A**ttention with **C**onvolutions **TCN** (NAC-TCN) which incorporates the benefits of attention and Temporal Convolutional Networks while ensuring that causal relationships are understood which results in a reduction in computation and memory cost. We accomplish this by introducing a causal version of Dilated Neighborhood Attention while incorporating it with convolutions. Our model achieves comparable, better, or *state-of-the-art* performance over TCNs, TCAN, LSTMs, and GRUs while requiring fewer parameters on standard emotion recognition datasets. We publish our code online for easy reproducibility and use in other projects – Github Link.

CCS CONCEPTS

• **General and reference** → Experimentation; *Performance*; • **Human-centered computing** → Collaborative and social computing devices; • **Computer systems organization** → **Neural networks**; • **Computing methodologies** → *Scene understanding*; *Vision for robotics*; **Activity recognition and understanding**; **Computer vision tasks**; **Computer vision**; **Computer vision problems**; **Machine learning approaches**.

KEYWORDS

Temporal Convolutional Networks, Video Understanding, Recurrent Neural Networks, Attention-Based Video Models, Emotion Recognition, AFEW-VA, AffWild2, EmoReact

1 INTRODUCTION

The study of emotion recognition has gained significant importance due to its widespread applications in various disciplines, including Human-Computer Interaction. Socially assistive robots, medical diagnosis of disorders such as PTSD [50], and software usability testing [23] in particular rely heavily on accurate emotion recognition. For instance, a better understanding of human emotions enables socially assistive robots to comprehend and respond appropriately to various scenarios, leading to effective assistance [2]. As a result, the development of advanced emotion recognition techniques holds immense potential in revolutionizing various aspects of our lives.

A common issue with the task of emotion recognition and understanding is lack of data. This is commonly due to the fact that annotation is a non-trivial task. This means that architectures must be expressive enough without large data sizes. The classification of emotions from video inputs is a task that has been extensively studied in the field of computer vision. A common approach to this problem involves performing classification directly on individual frames using convolutional networks[49]. However, this approach ignores the temporal aspect of emotion, which is critical to its accurate recognition. Humans exhibit emotions over a period of time, and considering this temporal aspect can lead to significant improvements in performance through contextual understanding [46].

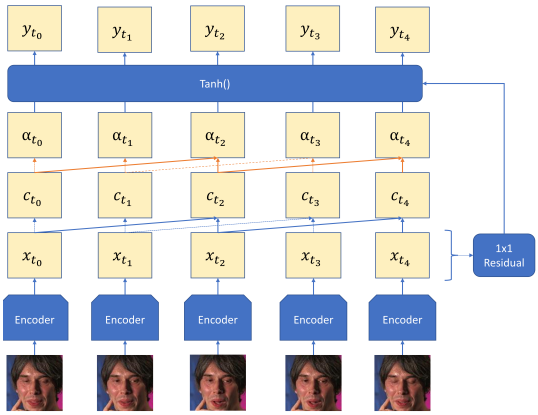


Figure 1: The NAC-TCN combines Dilated Temporal Convolutions with Dilated Neighborhood Attention to better capture temporal relationships in video inputs through contextual weighting using Dilated Neighborhood Attention. Our proposed architecture achieves better performance with smaller model size.

To incorporate temporal information into emotion recognition from video inputs, various techniques have been proposed, including Recurrent Neural Networks (RNNs) [48] and Transformers [8, 16, 61]. While GRUs and LSTMs are effective in modeling temporal dependencies, they suffer from slow training times, unstable training, and high computational costs. This is because their gradients flow through time rather than in parallel[22]. Transformers, on the other hand, typically suffer have a large parameter size and require more operations due to their self-attention mechanism attending to a large receptive field. Commonly the two models are combined for optimal performance[58].

An alternative solution that has gained popularity is the Temporal Convolutional Network (TCN) [5, 44, 59]. The TCN allows for the modeling of temporal dependencies in video tasks and time-series

tasks similar to RNNs and LSTMs, but with more stable gradients and higher efficiency at large receptive fields due to parallelized computation of gradients [5, 34]. The receptive field of the TCN can be easily adjusted with the number of layers, kernel size, and dilation factor. The TCN is a promising method for video tasks due to its efficiency benefits while understanding temporal dependencies [3] for emotional understanding.

Temporal Convolutional Networks do pose performance issues in regard to it's understanding complex relationships in the short and long term of a sequence and more irregular sequences [40]. Models attempt to address this, but fall short in terms of model size and performance due to the large size of self attention or auxiliary models [6, 40, 61]. TCAN attempts to intertwine attention layers [17] which leads to larger models.

In this paper we introduce a new Temporal Convolutional Network known as NAC-TCN which addresses concerns about complex temporal relationships while maintaining or improving the benefits of TCNs – stable gradients and computational efficiency.

1.1 Related Works

1.1.1 Recurrent Networks. Long-Short Term Memory Units (LSTMs) are a response to the hard to train nature of RNNs [45]. RNNs store a hidden state that can be used in order to represent all prior knowledge. Though this may be a strong representation, there are often training issues in long sequences. LSTMs include a forget gate that can allow for data to not be stored in the cells memory unit state. The LSTM has an input, output, and forget gate. LSTM models suffer from a large amount of operations per cell. Gated Recurrent Units [9] attempt to solve this by not holding a memory unit, instead only having update and forget gates. Despite the less complex structure, it has similar performance to LSTMs [11].

1.1.2 Attention and Transformer. Self Attention [9, 54] was introduced for language tasks. Instead of simply holding a weight and bias, self attention focuses on parts of the input and weights sequential inputs based on a query, key, and value matrix. These attention models can be used in conjunction with recurrent models for better performance (Sec. 1.1.1).

Transformers use layered attention with encoders and decoders [12, 54]. This has been commonly used in NLP tasks, but recent advances have moved towards it's application in computer vision [4].

These 2 methods represent a divergence in machine learning – use of classical recurrent methods or a move to the more computationally heavy transformer. In this paper, we hope to show a third path that can incorporate the benefits of both while alleviating their pitfalls.

2 BACKGROUND

2.1 Temporal Convolutional Networks

The Temporal Convolutional Network [5, 44, 59] is a convolutional representation of temporal data. It contains 2 commonly used parts, the main casual convolution network (Sec. 2.1.2) and dilated convolution (Sec. 2.1.1) in order to create a Dilated Temporal Convolutional Network.

2.1.1 Dilated Convolution. The dilation (à trous Convolution) [59] allows for a model to have a larger receptive field without increasing parameters. Dilated convolution is achieved by introducing "holes" between the points addressed by the kernel, resulting in a larger receptive field. The term "gaps" will be used to refer to any method of expanding a kernel through gaps to widen the receptive field.

2.1.2 Dilated Temporal Convolutional Network. Introduced in WaveNet [44], a Dilated Temporal Convolutional Network is a temporal network model that computes timesteps in parallel rather than sequentially. This fundamentally alters how the model addresses backpropagation through time by performing backpropogation for all time steps at once rather than following a temporal gradient flow. A casual convolution is used in order to prevent leakage from the past into future steps.

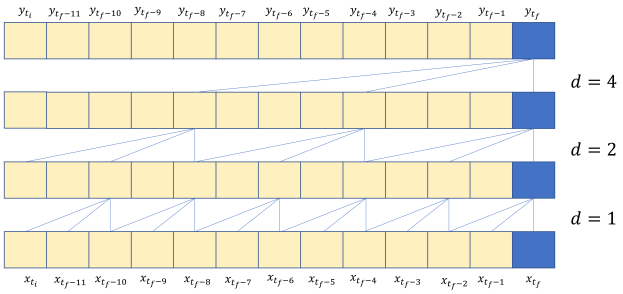


Figure 2: TCN architecture

A TCN layer can be described as follows for an input sequence x, dilation d, length i, dilated convolution $*_d$, and filter f

$$F(x_t) = (x *_d f)(t) = \sum_{n=0}^{i-1} f(t) \cdot x_{s-d*i}. \tag{1}$$

A dilated convolution is used in order to allow a network to understand time steps from previous steps efficiently and exponentially increase the receptive field [59]. Without dilated convolutions, TCNs would have a linear receptive field to prior steps. With dilations, the receptive field to previous timesteps (frames) can be calculated as

$$RF(n, d, k) = 1 + \sum_{i=0}^{n-1} d^i(k-1), \tag{2}$$

where d is the dilation factor, k is the kernel size, and n is the number of hidden layers. Commonly, a dilation factor of 2 is used in order to achieve an exponential receptive field [5].

Dilated Temporal Convolutional Networks (TCNs) allow for a large amount of temporal data to be processed with low computation through a large receptive field. TCNs allow for parallel computation, a large receptive field, and helps avoid vanishing or exploding gradients due to its backpropagation not being parallel to the temporal sequence, but rather perpendicular. Dilated TCNs have achieved impressive results replicating the long-term memory understanding of other architectures like LSTMs and RNNs such as the copying memory task [5]. The TCN has also been adopted for action segmentation achieving state-of-the-art results on action

detection [35]. TCNs have also been explored in emotion analysis, achieving results above those of LSTMs and RNNs on emotion based tasks [61]. TCNs commonly consist of temporal blocks, which are made up of two convolutional layers that are stacked on top of each other. The purpose of stacking these layers is to ensure that the input data is first scaled to the expected size and then passed through a convolutional layer of the output size.

2.2 Neighborhood Attention

Neighborhood Attention (NA) is an attention method that utilizes a sliding window technique similar to a convolution which views the time series at incriments like a convolution instead of all at once such as self-attention. This is similar to methods such as the SWIN transformer [20, 38] but the main difference comes from how NA allows for overlapping sgements, a method showed to improve performance by ensuring translation equivariance over similar methods [19, 20]. NA was introduced in order to address poor efficiency of self-attention and sliding window techniques by using a tiled algorithm and efficient CUDA kernels published in the $\mathcal{N}ATTEN$ library [1].

Dilated Neighborhood Attention (DiNA). is a method introduced to further address the performance of attention [19]. This dilated transformer works similar to dilated (also known as à trous) convolutions [59]. This improves performance beyond Neighborhood attention by attending to a higher receptive field in less operations than a normal transformer. When d is a dilation value and k is a neighborhood (kernel) size, DiNA reduces time complexity of self-attention from $O(n^2d)$ to $O(ndk)$.

2.2.1 TCAN. TCAN [17] is a TCN-based model that intertwines attention to maintain receptive field while providing an attention mechanism. This model has seen improvements over the TCN on language datasets. Although the method has seen improvements over the TCN, it leads to a significant parameter increase and it doesn't preserve the casual nature of the TCN, meaning that information flows freely between layers, and loses the temporal property due to leakage.

3 METHODS

To enhance the representation of temporal dependencies and their importance in emotional understanding, we propose a method that extends TCNs by incorporating the attention features of Neighborhood Attention while maintaining causality. We introduce our proposed archetecture that achieves this along with memory and runtime benefits in this section.

3.1 NAC-TCN Formulation

The **N**eighborhood **A**ttention with feature extracting **C**onvolutions TCN (NAC-TCN), is a deep learning based approach that utilizes Dilated Neighborhood Attention to enforce causality and combines convolutional operations and self-attention. Our proposed NAC-TCN method incorporates neighborhood self-attention layers within Temporal Blocks with 1D Convolutional Layers to allow the TCN to identify the most important frames through Neighborhood Attention and create local filters through 1D Convolutions. A combination of convolution and attention layers has been shown to produce improved results [57]. Our method makes use of Dilated Neighborhood Attention [19, 20, 56], and shifting inputs to maintain causality. 1x1 convolutions [37] are added on the input of each temporal block in order to ensure that residual connections have the same tensor shape in a similar fashion to the original TCN. The use of Dilated Neighborhood Attention not only keeps causality in the TCN, but also reduces operations and parameters.

3.2 Temporal Block

A NAC-TCN temporal block is parametrized by its kernel size k, dilation value d, input x, time step t, convolution f_k and the DiNA operation (Eq. (6)) and can be described as

$$F(x_t) = (x *_d f_k *_d \mathrm{DiNA}_k^d)(t). \tag{3}$$

In between each convolution, an activation (ReLU) is applied and followed by a 1D Spatial Dropout Layer which allows for feature independence between channels of the model [52]. This reflects the primary diagram in Sec. 3 which shows the Temporal Block structure where a Convolution is followed by Dropout and ReLU with a 1x1 convolution as described in Sec. 3.4.

3.2.1 Motivation for Convolution and Neighborhood Attention Stacking. We wish to create high performing low operational cost models. Adding convolutions achieves this twofold: being able to reduce dimensionality through downsampling (a feature that doesn't exist in NAT) and using a convolution over an attention block with fewer parameters.

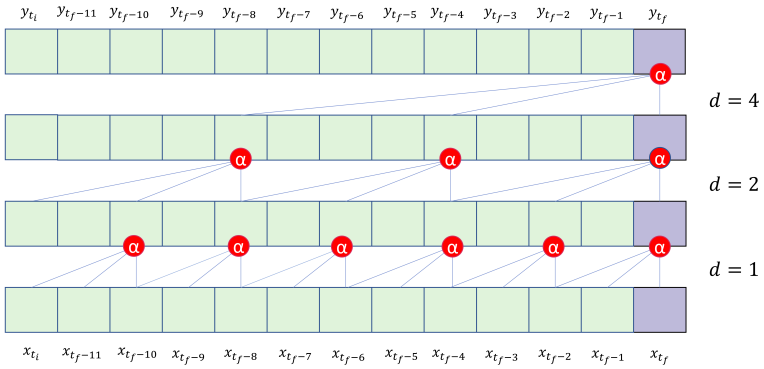
Additionally, our motivation for stacking convolutions and DiNAT comes from the benefits of convolutions that have been understated by recent works. Prior work has noted that though attention based methods outperform, they mainly do on a very large scale [14]. Primarily, lack of convolutions loses the benefit of quick and easy translational equivariance and requiring larger datasets/training time to perform as expected or a use of regularization [53]. This becomes important in domains such as emotion recognition, as datasets are tedious to collect as they require expert annotators and a mix of annotators requires agreement in labeling, which can be sometimes subjective[1].

3.3 Causal Dilated Neighborhood Attention

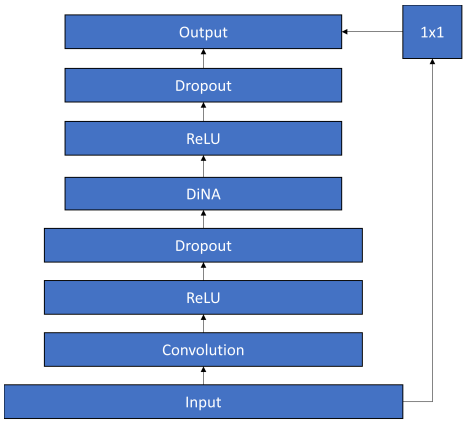
We extend the dilated neighborhood attention structure introduced by Hassani and Shi [19] (Sec. 2.2). Their Dilated Neighborhood Attention is modeled by attention weights $\mathbf{A}_i^{(k,\delta)}$ for a dilated by δ DiNA layer

$$\mathbf{A}_i^{(k,\delta)} = \begin{bmatrix} Q_i K^T_{\rho_1^\delta(i)} + B_{(i,\rho_1^\delta(i))} \\ Q_i K^T_{\rho_2^\delta(i)} + B_{(i,\rho_2^\delta(i))} \\ \vdots \\ Q_i K^T_{\rho_k^\delta(i)} + B_{(i,\rho_k^\delta(i))} \end{bmatrix}, \tag{4}$$

[1] Kollias [29] notes that many frames get thrown away due to annotator disagreement, making emotion datasets sparse and small. Additionally, the process itself is much more tedious than common classification datasets.



(a) α represents the Attention. Convolutions and Attention for timesteps other than t_f are omitted for simplicity.



(b) NAC-TCN Temporal Block

where $\rho_j^\delta(i)$ is i's j-th nearest neighbor, $B_{(i,\rho_k^\delta(i))}$ is bias corresponding to two tokens i and j, Q and K are query and key projections of X. The neighboring values of each neighborhood of size k dilated by a value δ is

$$\mathbf{V}_i^{(k,\delta)} = \begin{bmatrix} V^T_{\rho_1^\delta(i)} & V^T_{\rho_2^\delta(i)} & \cdots & V^T_{\rho_k^\delta(i)} \end{bmatrix}^T. \tag{5}$$

For each sliding window, we can define the output of each pixel by

$$\text{DiNA}_k^\delta(i) = softmax\left(\frac{\mathbf{A}_i^{(k,\delta)}}{\sqrt{d_k}}\right)\mathbf{V}_i^{(k,\delta)}, \tag{6}$$

where $DiNA$ is applied to each element i, where $i \in \mathbb{R}^{1\times n}$, and the output is then $DiNA(i)$. V, Q, and K are all calculated in the same manner as self-attention, as $DiNA(i)$ tends to simply self-attention as you increase k and decrease d to 1.

In order to stop temporal leakage, we must ensure that for an input $(x_t|x_0, ..., x_{t-1})$ to each temporal block, the output $(y_t|y_0, ..., y_{t-1})$ must be influenced by a time step at least 1 less than t. To ensure this, we change $\rho_j^\delta(i)$ to represent the nearest neighbor to the left of the i-th value with a dilation δ. This would mean that for a neighborhood k, the farthest value referenced is $\delta \cdot (k-1)$ to the left of the input sequence (not including the x_t itself), rather than $\frac{\delta\cdot(k-1)}{2}$. In order to ensure this, we pad $\text{DiNA}_k^\delta(i)$ and the convolutions using casual padding in order to make sure that timesteps are not influenced by the future, then removing padding before the next temporal block to ensure length consistency. In implementation, this is a standard $\delta \cdot (k-1)$ zero padding followed by removing $\delta \cdot (k-1)$ elements to the right, removing future timesteps.

3.4 Residual Connections

Since a network requiring a large receptive field will require an increase in layers, a residual connection [21] is added to address vanishing and shattering gradients problems [7, 55, 60], improve the loss landscape [36] leading to more stable training and better results. Residual layers are simply described as

$$H(x) = F(x) + x. \tag{7}$$

Since Temporal Blocks commonly upscale or downscale inputs, the residual layer in the NAC-TCN Temporal Block is

$$H(x) = F(x) + G(x). \tag{8}$$

where $G(x)$ is an optional 1x1 convolution used when scaling of channels is required. The 1x1 convolution impact is twofold: reducing dimensions of the network in later layers and providing a way for the model to translate features from one layer to another while maintaining the same overall information as previous layers. DiNA is not used for this 1x1 convolution because of its inability perform dimension scaling.

NAC-TCN reduces parameters compared to the multi-model approach proposed by others [18, 62, 64], the original TCN, and TCAN [17]. This can be attributed to the fact that attention operations, which solely consider the kernel size of the neighborhood, are intertwined with the convolution operations, leading to a decrease in parameters when compared to traditional combined structures.

4 EXPERIMENTS

In order to evaluate the effectiveness of our TCN methods, we used a variety of emotion and action recognition datasets, where newer temporal information is more relevant than the past. The $regnet\_y\_400mf$ image encoder[47] is used as an encoder for all the datasets to ensure that the NAC-TCN is the main factor tested.

The AffWild2 dataset. [23–28, 30–32] supplies 1,500,000+ annotated video frames of the valence and arousal metric in 341 videos. A video length of 256 frames is used. Due to the fact that valence and arousal are between $[-1, 1]$, tanh is applied to the model output. The valence and arousal scores are evaluated and trained on the Concordance Correlation Coefficient (CCC) metric

$$CCC = \frac{2s_{xy}}{s_x^2 + s_y^2 + (\bar{x} - \bar{y})^2}, \tag{9}$$

where x and y are predictions and ground truths, $2s_x$ and $2s_y$ are variances, and $\bar{x}$ and $\bar{y}$ are the mean values.

The EmoReact dataset. [43] provides videos of children annotated for 8 different emotions. Sequence length of 128 is used. The classes are not mutually exclusive and imbalanced. A random sampler and

binary cross entropy are used to address these issues. The same CNN encoder model from the AffWild2 experiments is used. Evaluation is done using Area under the precision-recall curve (AUC-ROC) to follow similar methodology to prior studies. AUC-ROC tells us for different threshold how a model performs by ploting False Positive (FP) rate and True Positive (TP) rate and defining AUC-ROC as the area under this curve.

AFEW-VA. [13, 33] provides valence and arousal annotations to popular films. These annotations are integers ranging from [-10,10]. These are converted to mood labels to compare to prior works [41]. Accuracy is used as the evaluation metric. The sequence length is 32, as videos are much shorter compared to other datasets.

4.1 Model Testing Methodology

The baseline GRU and LSTM hyperparameters for the Affwild2 dataset are chosen to match the prior models tested on the dataset. Other models such as TCN [5] and TCAN [17] used the same hyperparameters as the large NAC-TCN. GRU and LSTM blocks are concatenated with attention based models in order to create an ensemble of the two for testing.

For each evaluated dataset, NAC-TCN was tested in two sizes. The larger size attempted to use similar hyperparameters to the GRU models while ensuring optimal receptive field (1) through k, d, and number of layers. The optimal receptive field for all models besides AFEW-VA and Affwild2 were the length of the sequence, as only the final item was annotated. AFEW-VA used the entire sequence length 32 and Affwild2 used 256 based on prior literature. The model smaller size still ensured the optimal receptive field, but attempted to be a equal to smaller size than the GRU and LSTM models through adjusting previously mentioned hyperparameters along with the number of channels for the convolutional layers. This approach allowed us to conduct comparative tests while highlighting the versatility of the NAC-TCN model in terms of memory and computational cost.

4.2 Implementation Details

We use an Adam Optimizer with a base learning of 0.001 alongside an annealing cosine scheduler. We use a batch size of 16 for the Affwild2, EmoReact, and AFEW-VA datasets and all models were trained for 10 epochs. AFEW and Affwild all used subject based k-fold cross validation to ensure that information leakage did not occur between testing and training. The validation dataset of AffWild2 was used as the evaluation set and kept separate from training data. The EmoReact dataset had preset train and test splits that were used to be in line with the performance of prior models. The best performing model was selected for each method. The same random seed value was selected to ensure reproducibility. The number of heads for DiNAT was selected using hyperparameter search ($\{2^n \mid 1 \leq n \leq 3\}$).

4.3 Metrics

In addition to the per-dataset metrics, both operations and parameters are recorded. Operations are measured in MACs, or Multiply-Accumulate operations[2]. Both of these were measured using the Pytorch FLOPS Counter [51]. MACs represent the common operation of

$$(XW) + B, \tag{10}$$

where X is the original input, W is a weight, and B is a bias. We commonly report MMac, or MegaMACs (10^6 MACs).

[2]Note that the FLOP operation is $2 \times MAC$

5 RESULTS

In this section, we report the performance of our proposed NAC-TCN architectures against the baseline GRU, LSTM, TCN, and attention models with both performance and efficiency. In addition, we compare against state-of-the-art models on the respective datasets where relevant.

5.1 Affwild2

Table 1: Results on Affwild2 Validation. All models reproduced besides competition provided baseline. Bold denotes indicates the highest performing model.

Method	CCC ↑	M Params ↓	MMac ↓
Dataset Baseline [24]	0.17	–	–
Attn.	0.20	0.97	124
TCN	0.41	13.95	3079
GRU [42]	0.42	4.66	597
LSTM	0.42	6.21	797
GRU/Attn. [42]	0.44	5.63	721
LSTM/Attn	0.44	6.79	911
TCAN	0.46	17.12	3700
Transformer [10, 54]	0.48	31.04	3970
NAC-TCN (sm)	**0.48**	**1.24**	**381**
NAC-TCN (lg)	**0.52**	10.12	3230

As reported in Tab. 1, our proposed NAC-TCN architecture outperforms the other temporal-based models, while using a smaller model size. Additionally, we achieve the highest performance at smaller model sizes. The NAC-TCN, achieved through either a simple single layer setup or with self-attention, exhibited higher performance than the base TCN, which indicates that NAC-TCN can better learn temporal representations with less memory.

It should be noted that superior performance has exhibited in recent studies. However, conducting a direct comparison is challenging due to the utilization of disparate datasets and the employment of multi-sensor methodologies during the training process. Notably, participants in the latest Affwild2 challenge have surpassed reported results through four-encoder model, which incorporates audio and image encoders [39]. Other participants have surpassed prior state-of-the-art results with use of linguistic models from extracted words [63]. For our purposes, we have achieved a state-of-the-art result in the chosen set of input modalities and encoder choice.

Table 2: Results on the EmoReact Dataset. Bold denotes indicates the highest performing model. Two variations of our proposed model architecture are reported, where sm indicates the smaller model size and lg is the larger model size.

Method	AUC ROC ↑	MParams ↓	MMac ↓	External Training Data	Audio	Video
Attn.	0.56	2.5	360			✓
SVM [43]	0.62	–	–			✓
SVM [43]	0.63	–	–		✓	✓
GRU	0.74	1.62	208			✓
LSTM	0.75	2.16	277			✓
GRU/Attn.	0.76	4.4	567		✓	
LSTM /Attn.	0.76	4.9	636		✓	
TCN	0.79	1.8	459			✓
TSN [15]	0.79	–	–	✓		✓
TCAN	0.84	1.94	488			✓
NAC-TCN (sm)	**0.86**	**1.5**	453			✓
NAC-TCN (lg)	0.78	7.8	2500			✓

5.2 EmoReact

Results on EmoReact [43] (Tab. 2) show that with less modalities, NAC-TCN Small outperforms other models *without* multiple modalities or increased training data. This is done with a *decrease* in parameters and operations. This indicates that a better performing architecture like NAC-TCN may be actually outperform even with less data. NAC-TCN may be more prone to overfitting, given that with similar parameters to GRU and LSTM, it performed similarly and worse to a larger TCN. This highlights that NAC-TCN can be more expressive with the same hyperparamters, hence strong of the smaller model.

5.3 AFEW-VA

Table 3: Results on AFEW-VA dataset mood labels. Note that 1-CNN used a student teacher learning paradigm which may increase runtime beyond what is expected.

Method	Accuracy ↑	K Params ↓	MMac ↓
Attn.	0.43 ± 0.12	1860	234
1-CNN [41]	0.70 ± 0.10	–	–
TS (Mood/Δ)[41]	0.73 ± 0.08	–	–
TCN	0.74 ± 0.05	1220	99
GRU	0.75 ± 0.05	374	10
LSTM	0.75 ± 0.04	423	13
TCAN	0.75 ± 0.06	1253	120
LSTM/Attn	0.75 ± 0.46	1260	40
GRU/Attn	**0.76 ± 0.14**	1150	36
NAC-TCN (lg)	**0.76 ± 0.12**	988	84
NAC-TCN (sm)	0.75 ± 0.09	204	17

AFEW-VA results show that the larger NAC-TCN is able to outperform other methods. The smaller model results in similar performance to TCNs but with smaller memory footprint. It is important to note that the disparity between the models is minimal, within a ± 2% range. Consequently, the AFEW-VA dataset should be regarded primarily as a validation of the NAC-TCN's capacity to maintain performance levels akin to those of more expansive models. Nevertheless, NAC-TCN outperforms the 1-CNN model which uses attention [41].

5.4 Ablation Studies

In order to understand the impact of different choices we made in design and experimentation, we perform several ablation studies.

Table 4: Ablation study comparing residual connection on NAC-TCN small on the Affwild2 dataset.

CCC ↑	Residual
0.41	✗
0.48	✓

5.4.1 Residual Connection. We conduct on ablation study on the NAC-TCN Small model with the AffWild2 dataset, since the dataset has the deepest model due to the larger receptive field.

The study reveals that the residual connection is critical to training. Without residual connection, the model saw a significant performance decrease (Tab. 4).

Table 5: Small model used for NAC-TCN. Recall to Tab. 2 and Tab. 1.

Model	Causal	Affwild2	EmoReact
		CCC	AOC-ROC
NAC-TCN	✓	0.48	0.86
NAC-TCN	✗	0.44	0.65

5.4.2 Importance of Casuality. We compare the robustness to causality of our model versus other similar models in Tab. 5. Our model weights attention and applies convolutions based on previous k timesteps, where an acausal model would weight based on $\frac{k}{2}$ on each side. We find that the causal relation is important in both

datasets, but is more dramatic in the EmoReact dataset. This suggests that emotions are better learned when future information is unknown. This phenomenon of better learning with less information can be attributed to two potential reasons. Firstly, emotions inherently involve a causal process, wherein per-frame annotations occur continuously, thereby influencing annotators' decisions based on prior frames rather than knowledge of future frames. This can lead to different understandings depending on what context is used. Secondly, the disparity between the datasets stems from the variation in label format. Affwild2 employs per-frame labels, allowing for non-causal predictions of adjacent frames, whereas EmoReact utilizes end-of-video labels, thereby elevating the significance of causality (the last frame culminating in information from previous frames rather than $h/2$ prior frames). We find that prior literature commonly uses causal relationships over acausal with better results, making it an interesting point of discussion for future work.

6 DISCUSSION

6.1 Limitations

Although our method outperformed on many datasets, performance on AFEW-VA is notably similar to other temporal models. Given AFEW-VA is a smaller dataset, this may indicate that NAC-TCN outperforms other models on larger datasets with more oracle access. Multi-model & pretraining approaches that could perform better were not studied due to hardware limitations and simplicity in results. Our model also holds many of the same flaws of modern TCN based methods, such as higher memory during evaluation (needing the whole sequence instead of hidden state) and poor transfer learning with different k or d values.

6.2 Contribution

In this paper, we presented an alternative to the Temporal Convolutional Network that allows for attention while *decreasing parameters and number of MAC operations*. Experimental evaluation revealed improvements over classical methods such as GRUs, LSTMs, and Attention-based methods at a *lower computational cost*. Our method outperforms common temporal methods, improves on the benefits of the TCN, and performs similarly at an efficiency benefit while maintaining the same TCN controls over memory usage.

REFERENCES

[1] Natten – neighborhood attention extension. https://github.com/SHI-Labs/NATTEN, 2023.

[2] Abdollahi, H., Mahoor, M., Zandie, R., Sewierski, J., and Qualls, S. Artificial emotional intelligence in socially assistive robots for older adults: A pilot study. *IEEE Transactions on Affective Computing* (2022), 1–1.

[3] Armandika, F., Djamal, E. C., Nugraha, F., and Kasyidi, F. Dynamic hand gesture recognition using temporal-stream convolutional neural networks. In *2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI)* (2020), pp. 132–136.

[4] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2021), pp. 6836–6846.

[5] Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271* (2018).

[6] Bai, S., Kolter, J. Z., and Koltun, V. Trellis networks for sequence modeling. In *International Conference on Learning Representations (ICLR)* (2019).

[7] Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question?, 2018.

[8] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers, 2020.

[9] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[10] Chudasama, V., Kar, P., Gudmalwar, A., Shah, N., Wasnik, P., and Onoe, N. M2fnet: Multi-modal fusion network for emotion recognition in conversation, 2022.

[11] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[12] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[13] Dhall, A., Goecke, R., Lucey, S., and Gedeon, T. Collecting large, richly annotated facial-expression databases from movies. *IEEE Multimedia 19* (09 2012), 34–31.

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[15] Efthymiou, N., Filntisis, P. P., Potamianos, G., and Maragos, P. Visual robotic perception system with incremental learning for child–robot interaction scenarios. *Technologies 9*, 4 (2021).

[16] Gupta, A., Arunachalam, S., and Balakrishnan, R. Deep self-attention network for facial emotion recognition. *Procedia Computer Science 171* (2020), 1527–1534. Third International Conference on Computing and Network Communications (CoCoNet'19).

[17] Hao, H., Wang, Y., Xia, Y., Zhao, J., and Shen, F. Temporal convolutional attention-based network for sequence modeling. *arXiv preprint arXiv:2002.12530* (2020).

[18] Hao, H., Wang, Y., Xia, Y., Zhao, J., and Shen, F. Temporal convolutional attention-based network for sequence modeling. *CoRR abs/2002.12530* (2020).

[19] Hassani, A., and Shi, H. Dilated neighborhood attention transformer.

[20] Hassani, A., Walton, S., Li, J., Li, S., and Shi, H. Neighborhood attention transformer.

[21] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

[22] Jozefowicz, R., Zaremba, W., and Sutskever, I. An empirical exploration of recurrent network architectures. In *International conference on machine learning* (2015), PMLR, pp. 2342–2350.

[23] Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W., and Wrobel, M. R. Emotion recognition and its applications. In *Human-Computer Systems Interaction: Backgrounds and Applications 3*. Springer, 2014, pp. 51–62.

[24] Kollias, D. Abaw: Valence-arousal estimation, expression recognition, action unit detection and multi-task learning challenges. *arXiv preprint arXiv:2202.10659* (2022).

[25] Kollias, D., Schulc, A., Hajiyev, E., and Zafeiriou, S. Analysing affective behavior in the first abaw 2020 competition. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, pp. 794–800.

[26] Kollias, D., Sharmanska, V., and Zafeiriou, S. Face behavior a la carte: Expressions, affect and action units in a single network. *arXiv preprint arXiv:1910.11111* (2019).

[27] Kollias, D., Sharmanska, V., and Zafeiriou, S. Distribution matching for heterogeneous multi-task learning: a large-scale face study. *arXiv preprint arXiv:2105.03790* (2021).

[28] Kollias, D., Tzirakis, P., Nicolaou, M. A., Papaioannou, A., Zhao, G., Schuller, B., Kotsia, I., and Zafeiriou, S. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision* (2019), 1–23.

[29] Kollias, D., and Zafeiriou, S. Aff-wild2: Extending the aff-wild database for affect recognition, 2019.

[30] Kollias, D., and Zafeiriou, S. Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface. *arXiv preprint arXiv:1910.04855* (2019).

[31] Kollias, D., and Zafeiriou, S. Affect analysis in-the-wild: Valence-arousal, expressions, action units and a unified framework. *arXiv preprint arXiv:2103.15792* (2021).

[32] Kollias, D., and Zafeiriou, S. Analysing affective behavior in the second abaw2 competition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 3652–3660.

[33] Kossaifi, J., Tzimiropoulos, G., Todorovic, S., and Pantic, M. Afew-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing 65* (02 2017).

[34] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. Temporal convolutional networks for action segmentation and detection. *CoRR abs/1611.05267* (2016).

[35] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 156–165.

[36] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets, 2018.

[37] Lin, M., Chen, Q., and Yan, S. Network in network, 2014.

[38] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[39] Meng, L., Liu, Y., Liu, X., Huang, Z., Jiang, W., Zhang, T., Deng, Y., Li, R., Wu, Y., Zhao, J., et al. Multi-modal emotion estimation for in-the-wild videos. *arXiv preprint arXiv:2203.13032* (2022).

[40] Moor, M., Horn, M., Rieck, B., Roqueiro, D., and Borgwardt, K. Early recognition of sepsis with gaussian process temporal convolutional networks and dynamic time warping. In *Machine Learning for Healthcare Conference* (2019), PMLR, pp. 2–26.

[41] Narayana, S., Subramanian, R., Radwan, I., and Goecke, R. Focus on change: Mood prediction by learning emotion changes via spatio-temporal attention, 2023.

[42] Nguyen, H.-H., Huynh, V.-T., and Kim, S.-H. An ensemble approach for facial behavior analysis in-the-wild video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2022), pp. 2512–2517.

[43] Nojavanasghari, B., Baltrušaitis, T., Hughes, C. E., and Morency, L.-P. Emoreact: a multimodal approach and dataset for recognizing emotional responses in children. In *Proceedings of the 18th acm international conference on multimodal interaction* (2016), pp. 137–144.

[44] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[45] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013* (11 2012).

[46] Poria, S., Majumder, N., Mihalcea, R., and Hovy, E. H. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *CoRR abs/1905.02947* (2019).

[47] Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 10425–10433.

[48] Rangulov, D., and Fahim, M. Emotion recognition on large video dataset based on convolutional feature extractor and recurrent neural network. In *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)* (2020), IEEE, pp. 14–20.

[49] Saroop, A., Ghugare, P., Mathamsetty, S., and Vasani, V. Facial emotion recognition: A multi-task approach using deep learning, 2021.

[50] Schultebraucks, K., Yadav, V., Shalev, A. Y., Bonanno, G. A., and Galatzer-Levy, I. R. Deep learning-based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood. *Psychological Medicine 52*, 5 (2022), 957–967.

[51] Sovrasov, V. Flops-counter pytorch. https://github.com/sovrasov/flops-counter.pytorch.git, 2021.

[52] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 648–656.

[53] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers and distillation through attention, 2021.

[54] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.

[55] Veit, A., Wilber, M., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks, 2016.

[56] Walton, S., Hassani, A., Xu, X., Wang, Z., and Shi, H. Stylenat: Giving each head a new perspective.

[57] Xu, W., Xu, Y., Chang, T., and Tu, Z. Co-scale conv-attentional image transformers, 2021.

[58] You, J., and Korhonen, J. Attention boosted deep networks for video classification. In *2020 IEEE International Conference on Image Processing (ICIP)* (2020), pp. 1761–1765.

[59] Yu, F., and Koltun, V. Multi-scale context aggregation by dilated convolutions, 2016.

[60] Zaeemzadeh, A., Rahnavard, N., and Shah, M. Norm-preservation: Why residual networks can become extremely deep?, 2020.

[61] Zhang, S., An, R., Ding, Y., and Guan, C. Continuous emotion recognition using visual-audio-linguistic information: A technical report for abaw3, 2022.

[62] Zhang, S., An, R., Ding, Y., and Guan, C. Continuous emotion recognition using visual-audio-linguistic information: A technical report for abaw3, 2022.

[63] Zhang, S., Zhao, Z., and Guan, C. Multimodal continuous emotion recognition: A technical report for abaw5. *arXiv preprint arXiv:2303.10335* (2023).

[64] Zhao, Y., Wang, D., Xu, B., and Zhang, T. Monaural speech dereverberation using temporal convolutional networks with self attention. *IEEE ACM Trans. Audio Speech Lang. Process. 28* (May 2020), 1598–1607.