TemporalMaxer Performance in the Face of Constraint: A Study in Temporal Action Localization

A Comprehensive Analysis on the Adaptability of TemporalMaxer in Resource-Scarce Environments

Author
Teodor-Gabriel Oprescu
T.Oprescu@student.tudelft.nl

Responsible Professor
Dr. Jan van Gemert

Supervisors
Robert-Jan Bruintjes
Atilla Lengyel
Ombretta Strafforello

01 Introduction

Temporal Action Localization (TAL) is the task of detecting specific actions within a video, alongside its start time and end time.

Main issues for TAL models:

- Requiring large datasets of labeled videos. Collecting and annotating is time-consuming & costly
- Computationally expensive
- Large training time

We will explore a TAL state-of-the-art (SOTA) model called TemporalMaxer.

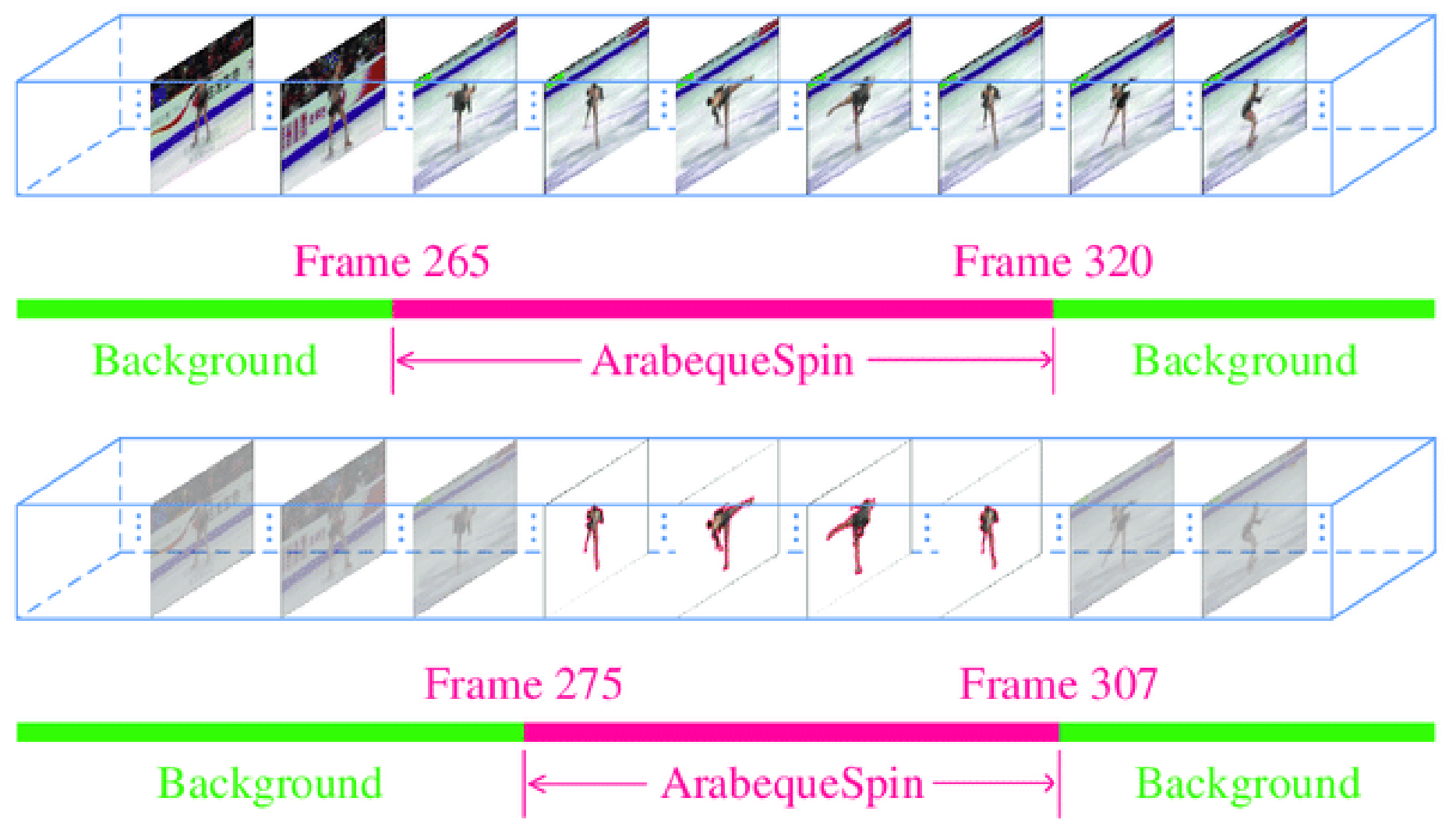

Figure 1: Illustration of TAL [1]

The Research question:
"How well does the TemporalMaxer method perform in a limited compute power and data setting?"

02 The Model: TemporalMaxer

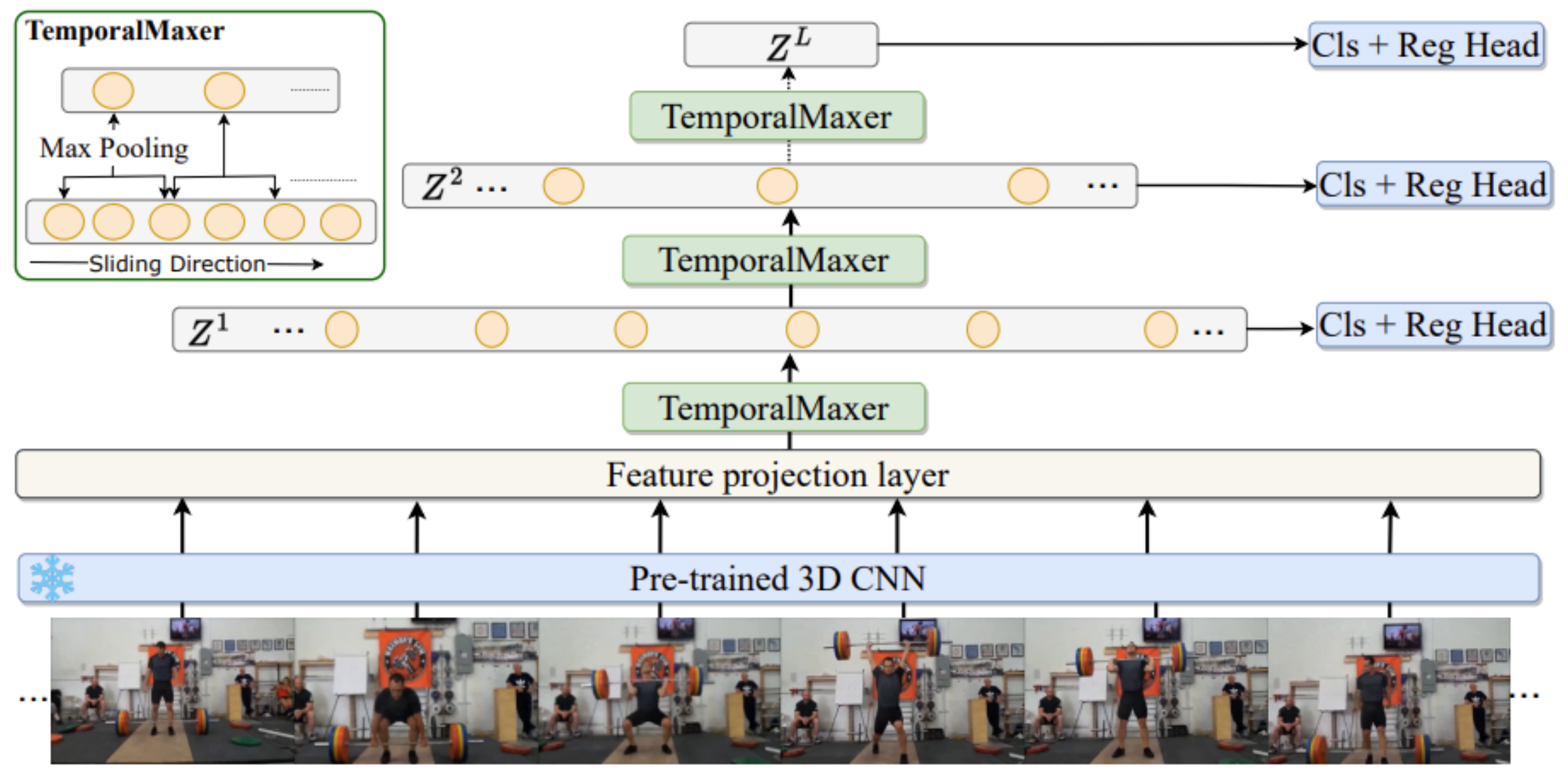

Figure 2: Overview of Temporal Maxer [2]

TemporalMaxer is inspired from another SOTA TAL model called ActionFormer. Its novel aspect is represented by the model's backbon based on a MaxPooling block → reduced complexity → fewer parameters & more computationally efficient.

03 Methodology

Data efficiency experiment:

Overview in **Algorithm 1 (Figure 3)**.

We train TemporalMaxer on increasingly-bigger parts of the THUMOS'14 [3] dataset (size p%).

We measure the performance of the model by testing it 5 times on the validation test for each **p** value. We use the mean average precision metric (mAP).

All experiments will be conducted on the THUMOS'14 [3] dataset.

Algorithm 1 Data efficiency evaluation procedure

```
$\mathcal{D} = \{(\mathbf{V_i}, \mathbf{y_i})\}_{i=1}^{N}$
$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \leftarrow \texttt{split}(\mathcal{D})$
for $p$ in $[10\%, 20\%, 40\%, 60\%, 80\%, 100\%]$ do
    mAPs $\leftarrow$ empty list
    for $i = 1, ..., 5$ do
        $\mathcal{D}_{\text{s}} \leftarrow \texttt{sample}(\mathcal{D}_{\text{train}}, p)$
        Train on $\mathcal{D}_{\text{s}}$
        mAP $\leftarrow \texttt{calculate-mAP}(\mathcal{D}_{\text{test}})$
        Append mAP to mAPs
    Report $\mu_{\text{mAPs}}$ and $\sigma_{\text{mAPs}}$
```

Figure 3: Algorithm for the evaluation of the data efficiency

Compute efficiency experiments:

- **Training**
  - Generate 5 random seeds:
    - For each seed, evaluate the model under normal conditions 5 times.
  - Report the mean and standard deviation of mAP and training time of all total 25 runs.
- **Inference**
  - For increasingly bigger sizes of input features:
    - Measure inference time, number of Multiply-Accumulate operations (MACs), memory usage, GPU utilization
  - Report the mean and standard deviation (if applicable) for all the metrics mentioned

04 Experiments

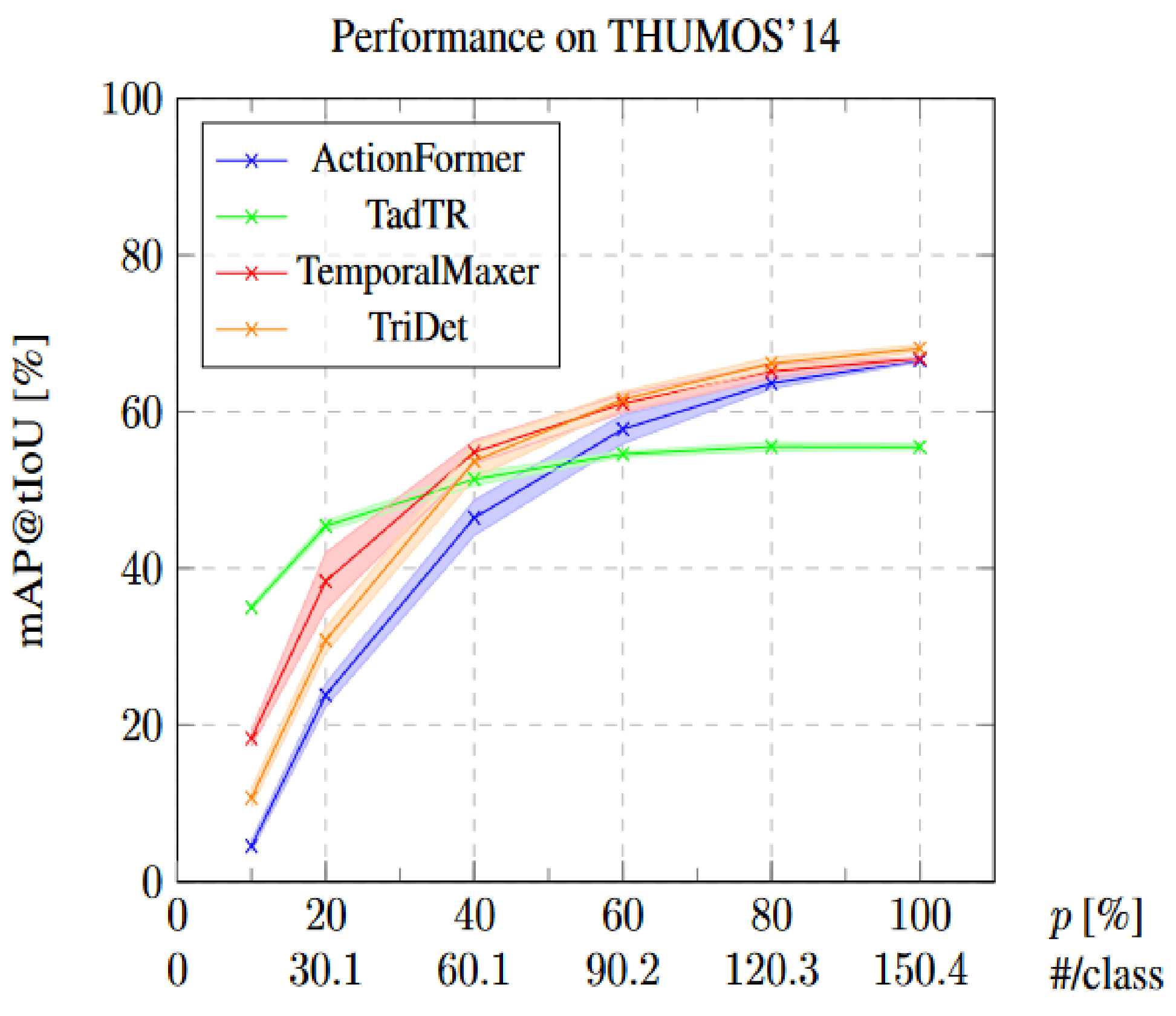

Figure 4: Performance of TemporalMaxer compared to other TAL models for the data efficiency experiment [4]

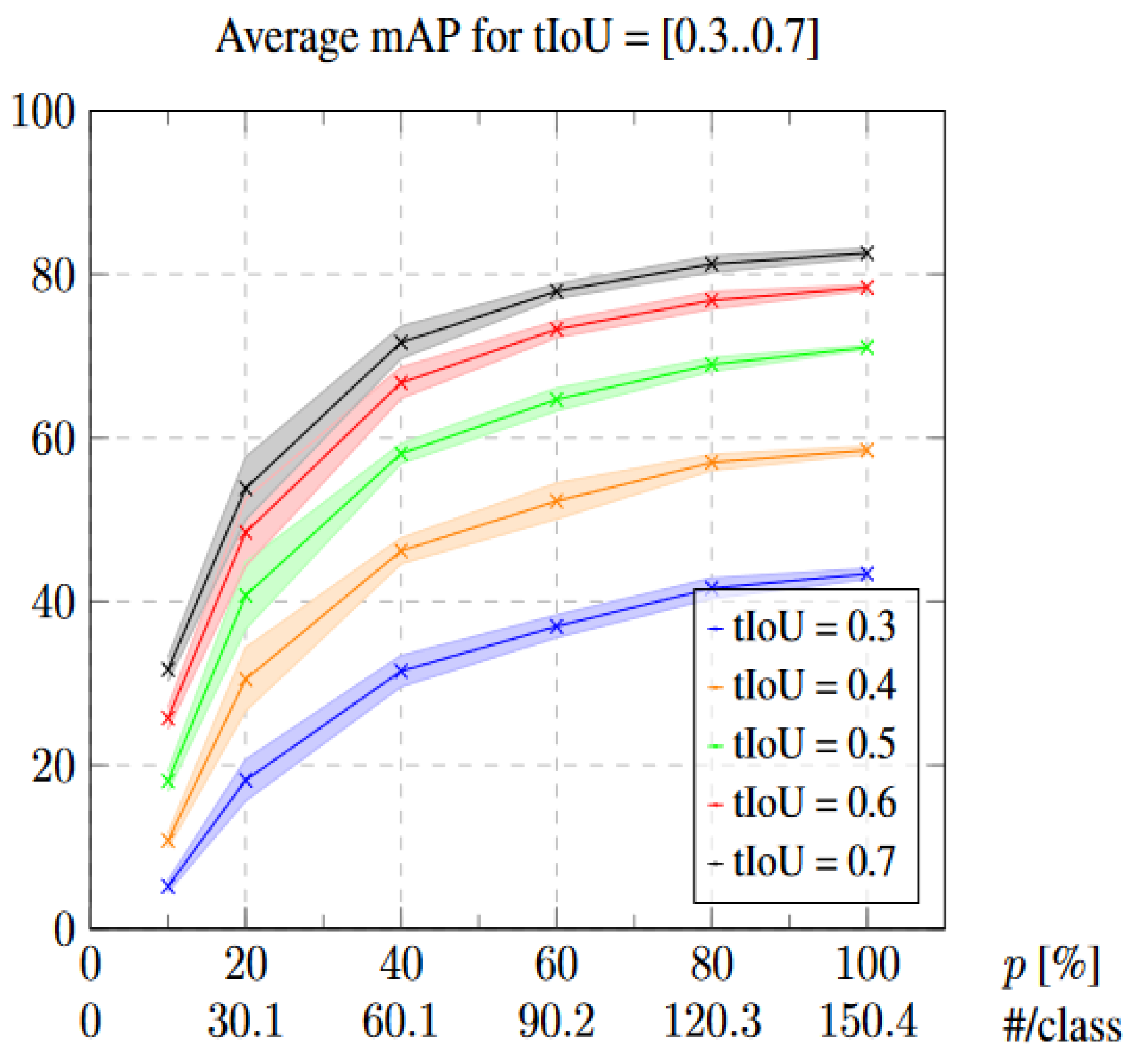

Figure 5: TemporalMaxer's data efficiency results for each tIoU value in the range of [0.3,0.7]

Model	Avg. mAP [%]	O. mAP [%]	Time [s]
TriDet [16]	68.07 ± 0.42	69.3	646.17 ± 26.12
TemporalMaxer [12]	66.96 ± 0.37	67.7	2955.64 ± 1659.98
ActionFormer [10]	66.5 ± 0.31	66.8	866.22 ± 26.97
TadTR [18]	55.3 ± 0.63	56.7	425.72 ± 3.469

Figure 6: TemporalMaxer's results for the training experiment compared to other TAL models

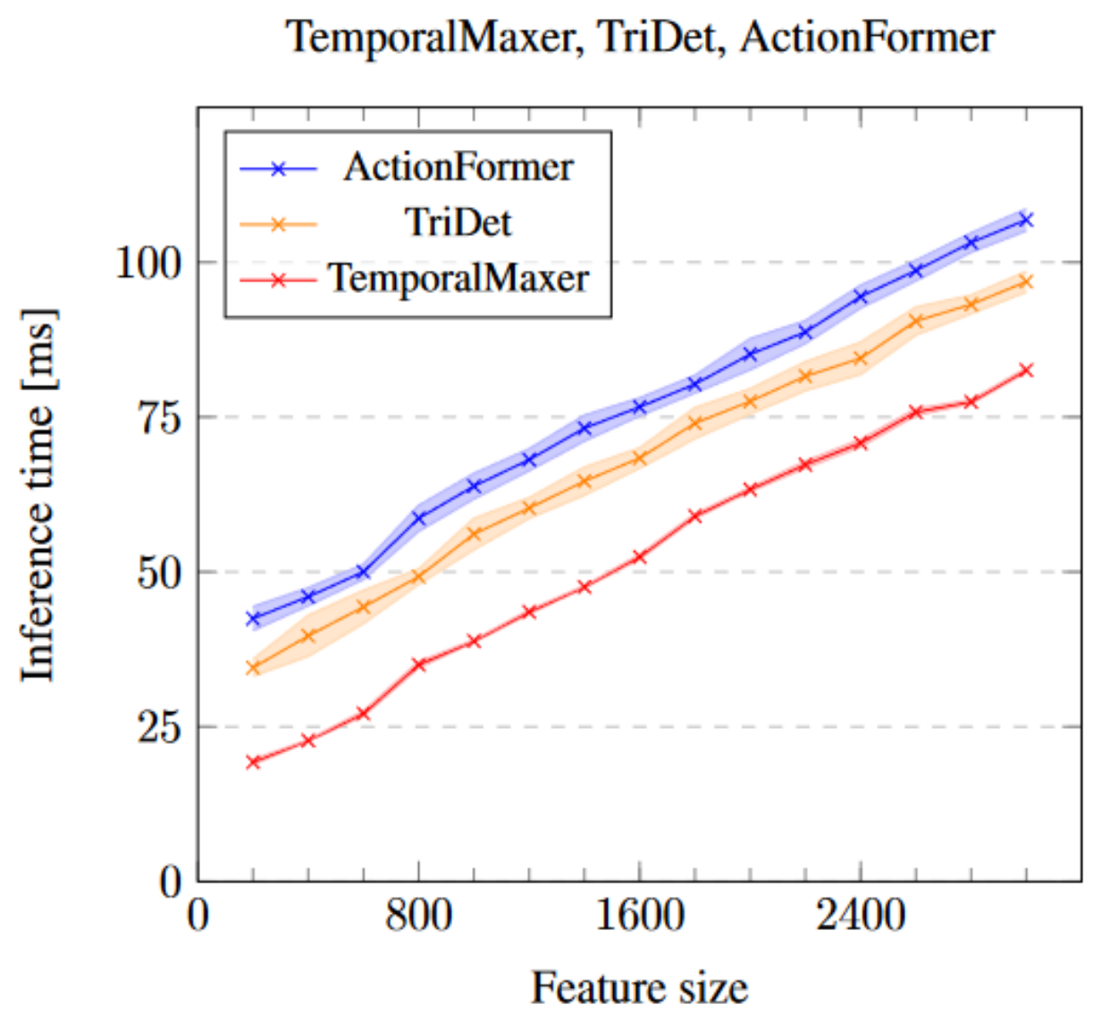

Figure 7: TemporalMaxer's Inference time compared to other SOTA TAL models

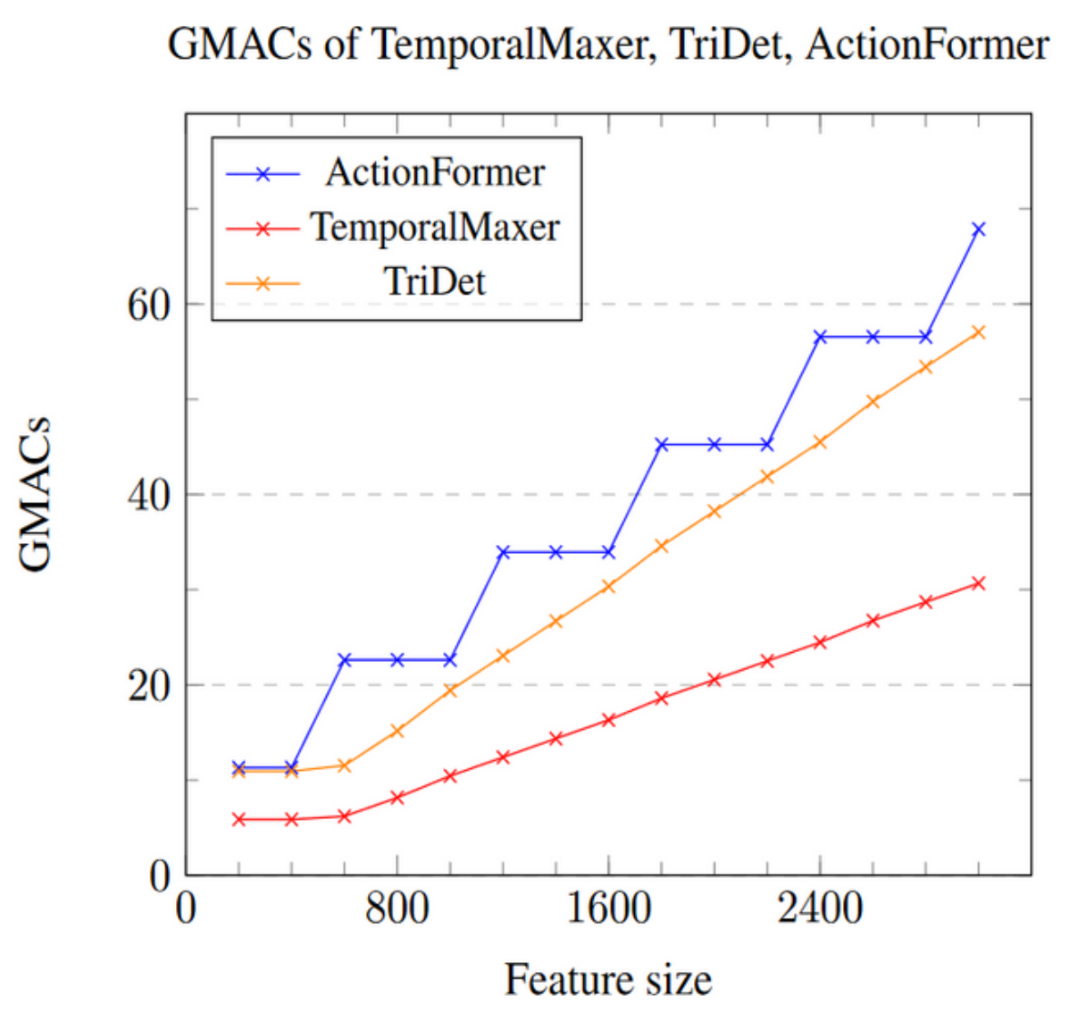

Figure 8: TemporalMaxer's GMACS compared to other SOTA TAL models

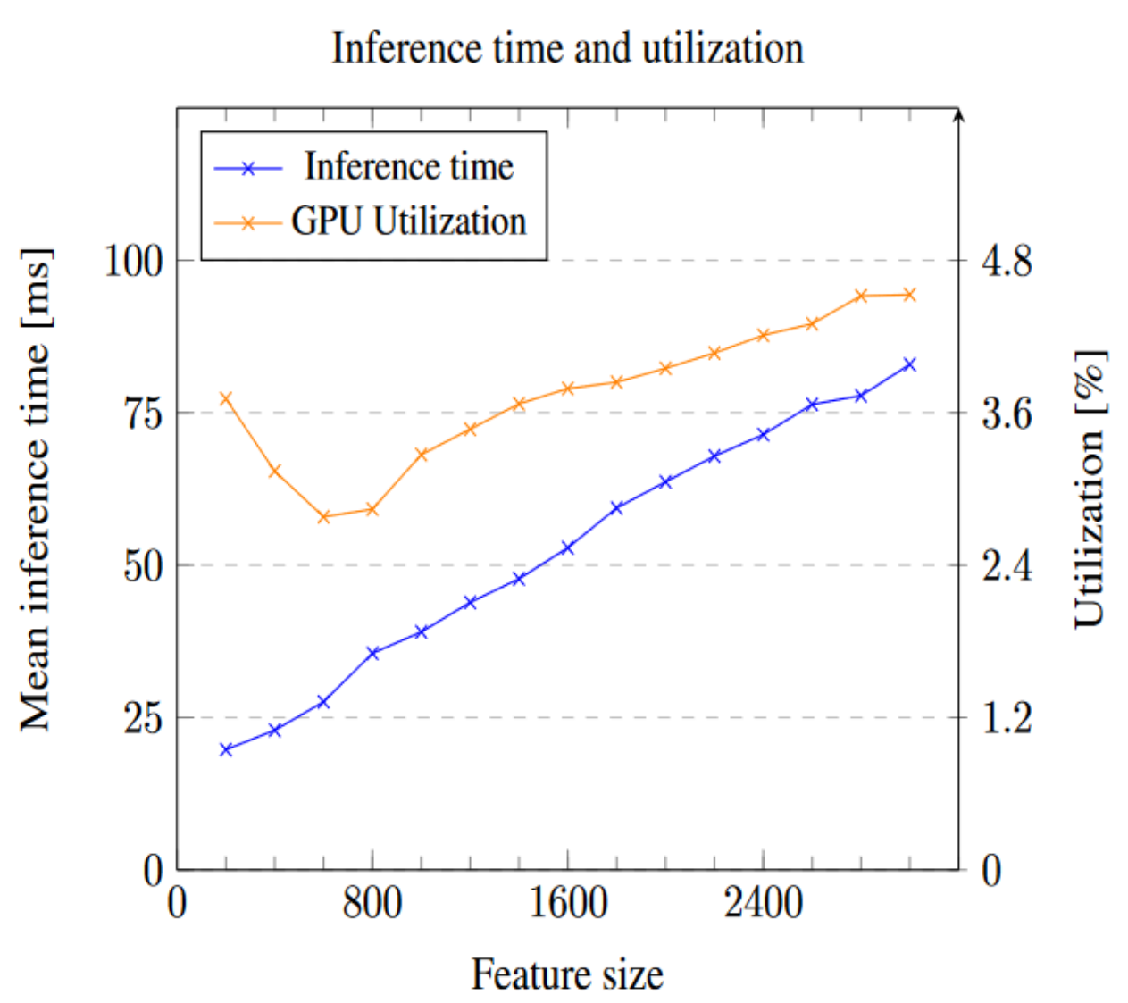

Figure 9: TemporalMaxer's inference time and GPU utilization

05 Conclusions

"A is more efficient in X than B" : A yields a better performance in X than B

Is TemporalMaxer ... compared to other SOTA TAL models?

- data efficient ✓
  - the model achieves significant performance with only 40-60% of the original training data
- training time efficient ✗?
  - the model presents an unusually high mean training time and standard deviation than other similar TAL models, more investigation needs to be done regarding this aspect
- computationally efficient (GMACs & inference time) ✓
  - TemporalMaxer's results indeed show that all the compute metrics increase linearly with the size of the input features. Moreover, TemporalMaxer significantly outclasses other similiar-in-performance TAL models on compute metrics

References

[1] Le Wang, Xuhuan Duan, Qilin Zhang, Zhenxing Id, Gang Hua, and Nanning Zheng. Segment-tube: Spatio-temporal action localization in untrimmed videos with per-frame segmentation. Sensors, 18, 05 2018.
[2] Tuan N Tang, Kwonyoung Kim, and Kwanghoon Sohn. Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. arXiv preprint arXiv:2303.09055, 2023.
[3] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

Acknowledgements

[4] Many thanks to **Jan Warchocki** for sharing the code necessary to generate the Figure 4 plot