# CSTA: CNN-based Spatiotemporal Attention for Video Summarization

Jaewon Son, Jaehun Park, Kwangsu Kim*
SungKyunKwan University
Applied AI & Computer Vision Lab

Poster: THU-JUN-20 17:00 ~ 19:00
Paper: https://arxiv.org/abs/2405.11905
Code: https://github.com/thswodnjs3/CSTA

# Task: Video summarization

**Train models to summarize long videos like the way humans do**

*Long original video "Cooking"*

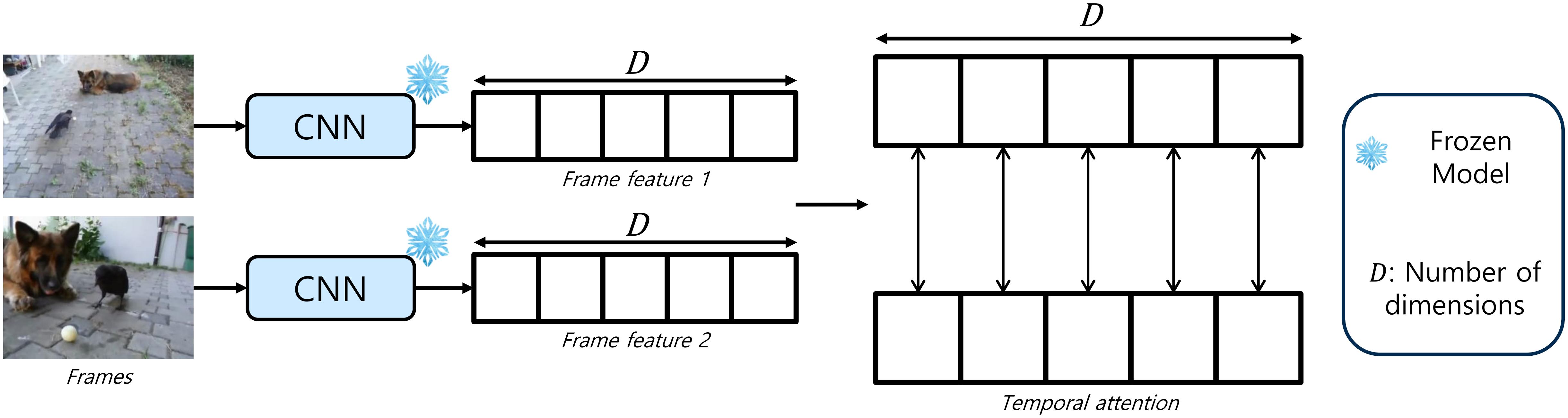
**Model**

*Short summarized video by UnpairedVSN(2019, CVPR)*

1. Models take long videos and are <u>trained to understand which frames are important</u>.
2. Based on decisions about keyframes, models generate summarized videos.
3. For better summarization, attention is commonly used to give weights for keyframes.

# Preliminary: Temporal attention

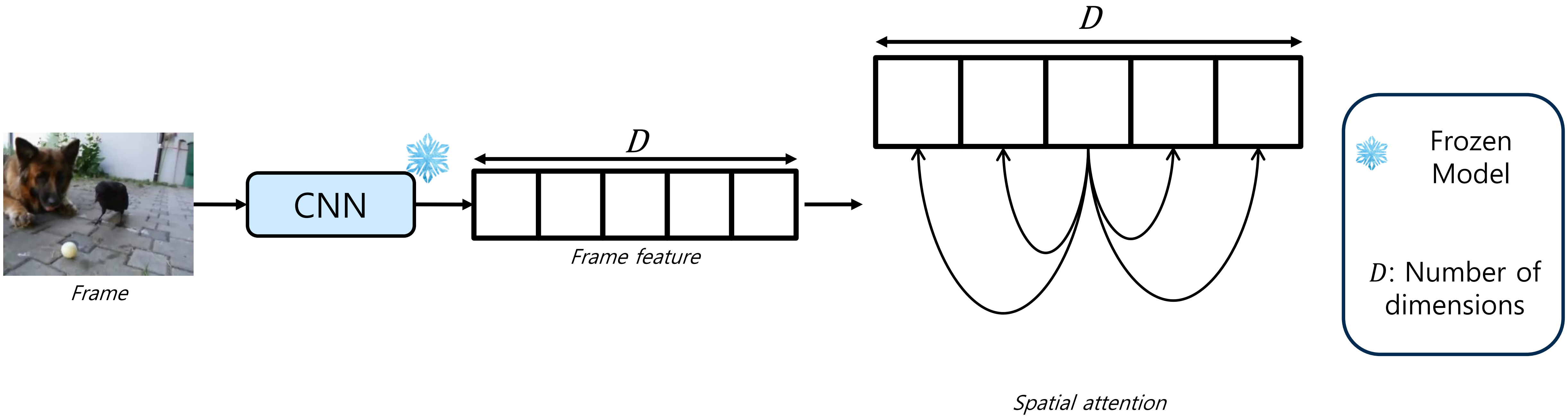
**Attention is based on cross-correlations of each attribute between features**




1. Frozen CNN models extract frame features.
2. Calculate attention by <u>cross-correlations of each attribute between pairs of features</u>.
3. Because of correlations between different frames, it is called temporal attention.

# Problem: Temporal attention lacks spatial weights

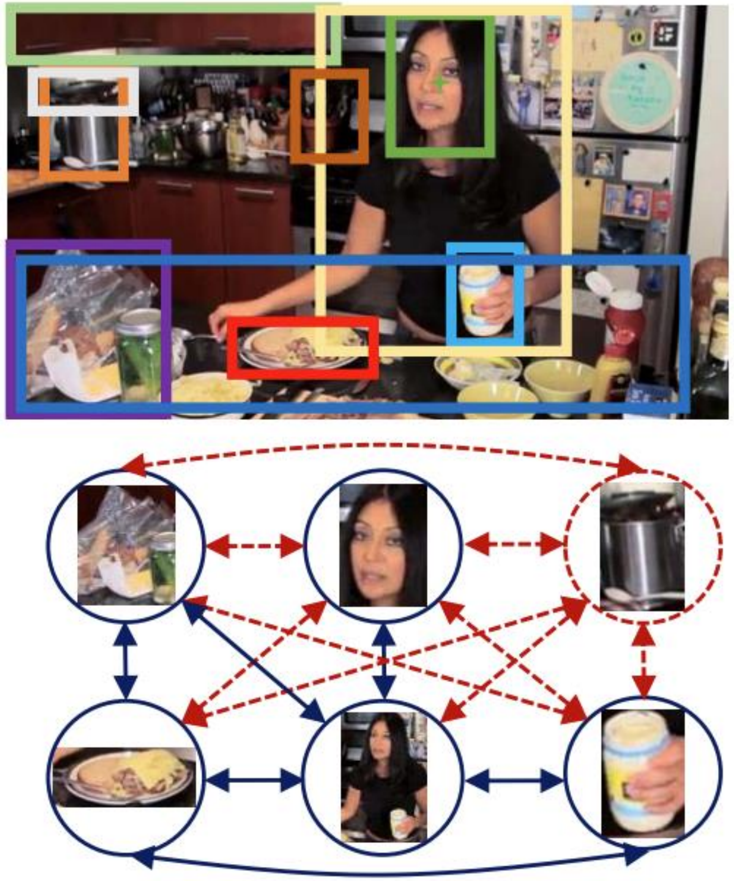
## The importance of attributes in a feature differs from temporal attention



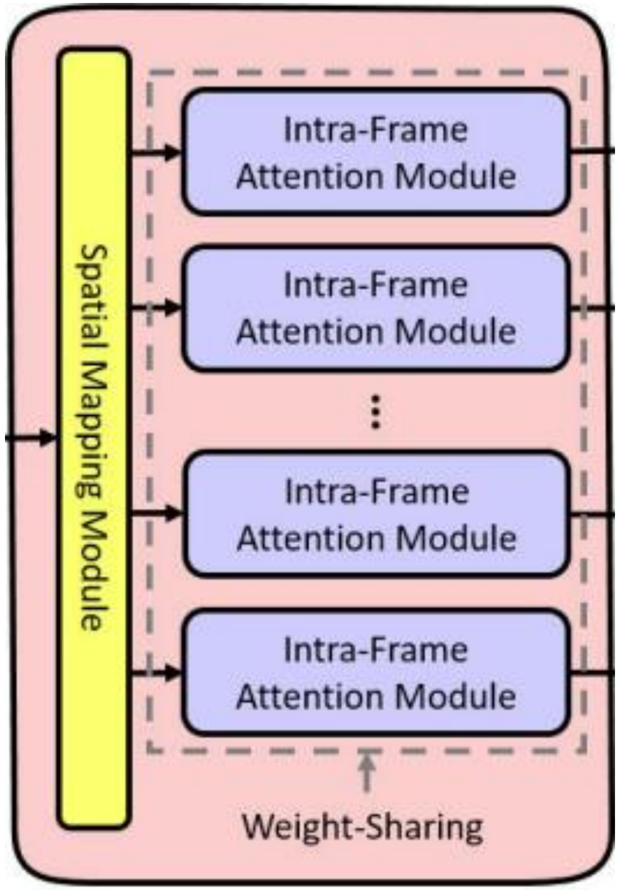

1. Each attribute of features indicates the visual characteristics of frames.
2. All attributes have their own importance in the frame.
3. Reflecting spatial attention changes the weights of attributes.
4. Temporal attention changes based on the spatial attention.
5. For precise attention, considering both temporal and spatial attention is necessary.

# Related work: Spatiotemporal attention methods

**Considering spatiotemporal attention requires huge costs for better results**



*RR-STG(2022, ITIP)*




*STVT(2023, ITIP)*

1. Previous works employ an extra model to include spatial attention. (e.g. object detection, self-attention)

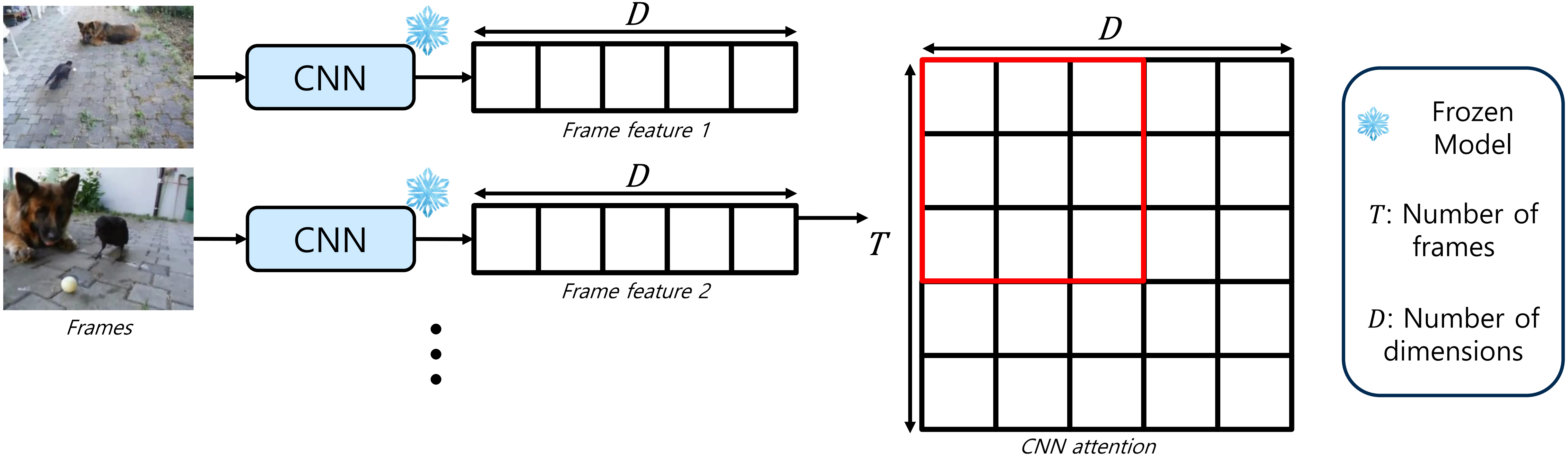
2. Using spatiotemporal attention performs better than temporal one only.

3. Processing every frame by additional model is very costly due to the long length of videos.

Goal: Video summarization models considering spatiotemporal attention and efficiency.

# Approach: CNN attention

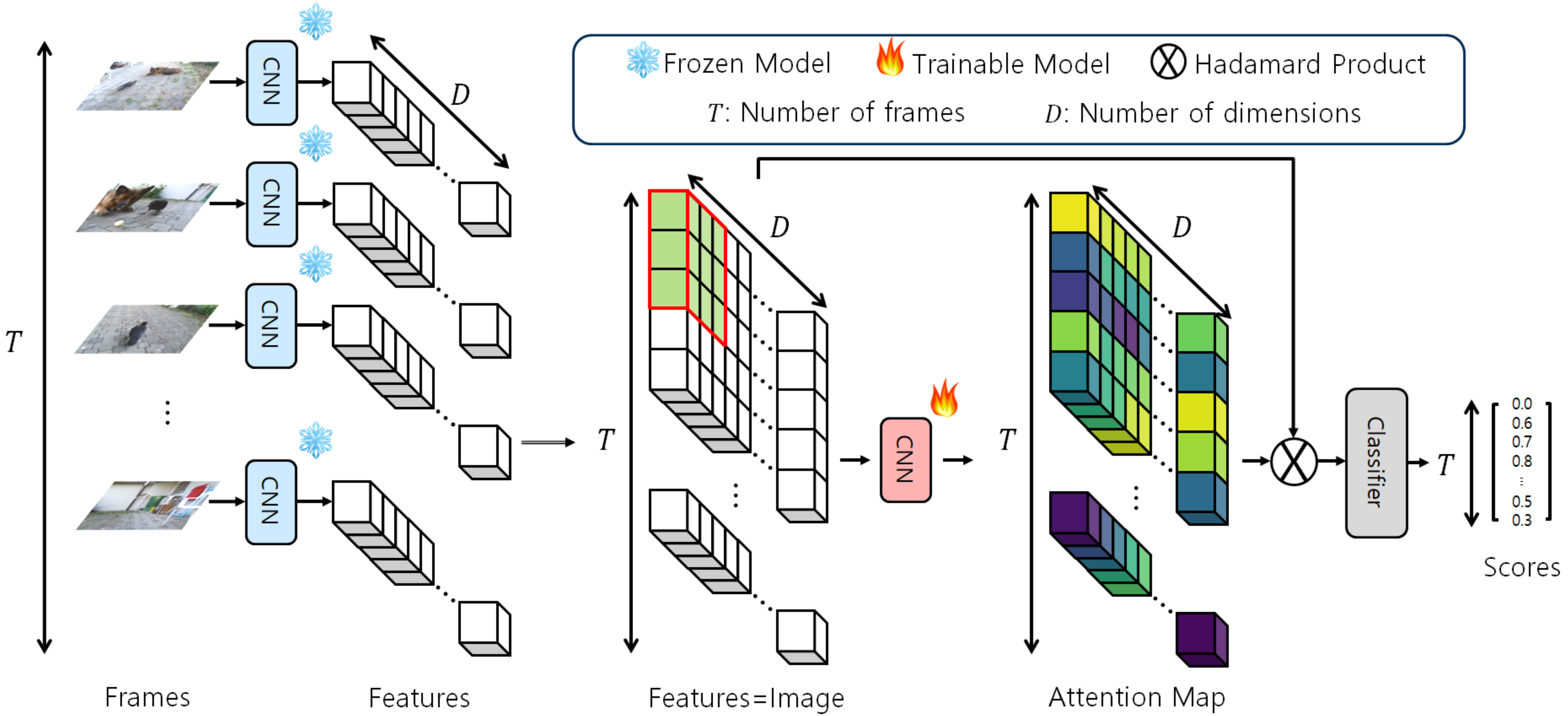
## Use CNN as attention for spatiotemporal weights and efficiency




1. Stack all frame features to form image-like frame features.
2. Consider features as images, and apply 2D CNN models to features for spatiotemporal attention.
3. CNN can work as attention due to its ability to learn the absolute positions of images.
   -PosENet(2020, ICLR), CPVT(2023, ICLR)
4. CNN reduces computations for attention which works in a pairwise way.
   -CeiT(2021, ICCV), CvT(2021, ICCV), CmT(2022, CVPR)

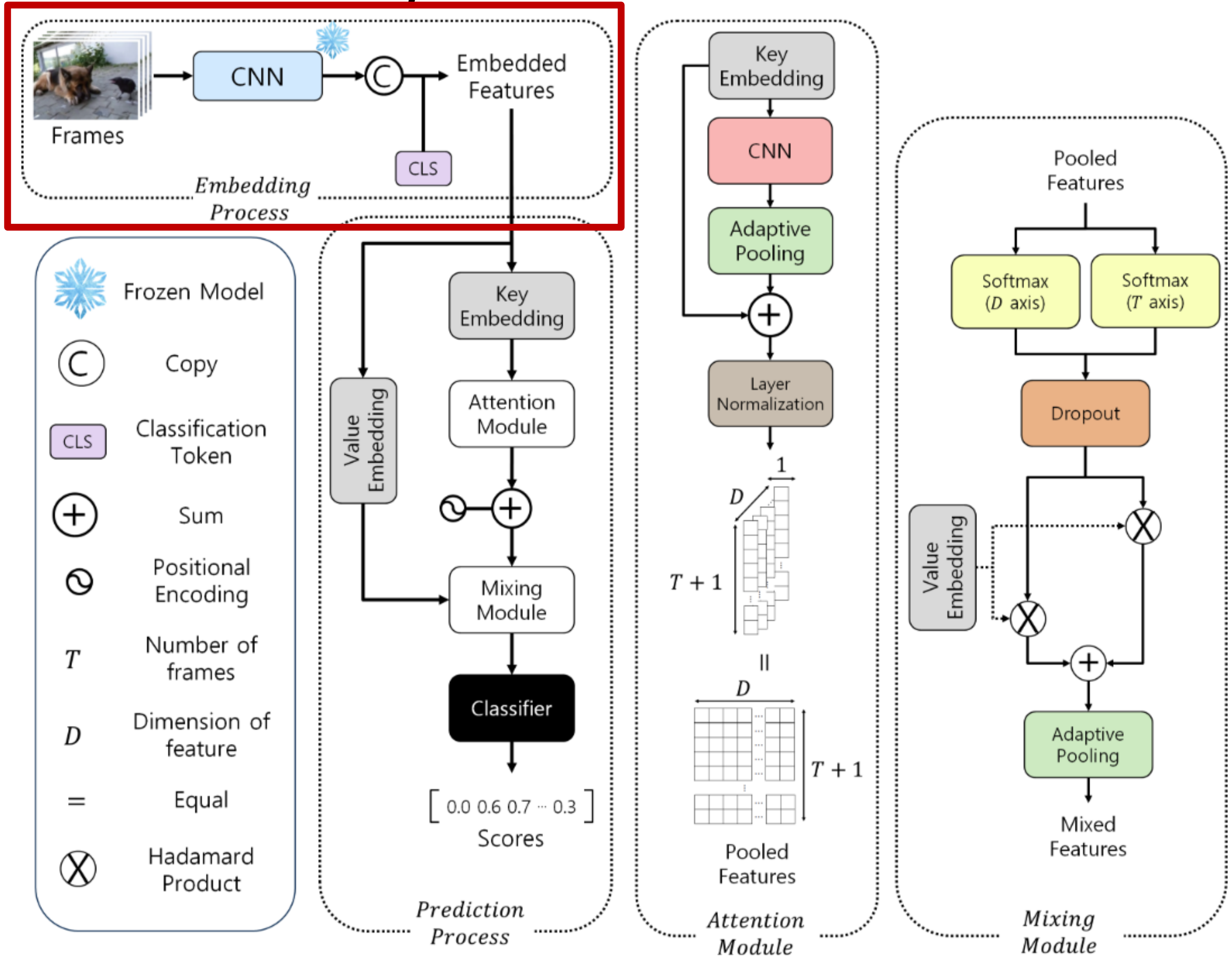
# Overview: (CSTA) CNN-based spatiotemporal attention

## 2D CNN creates attention maps by considering frame features as images



# Architecture: Embedding Process

## From frames, extract Embedded Features used as inputs



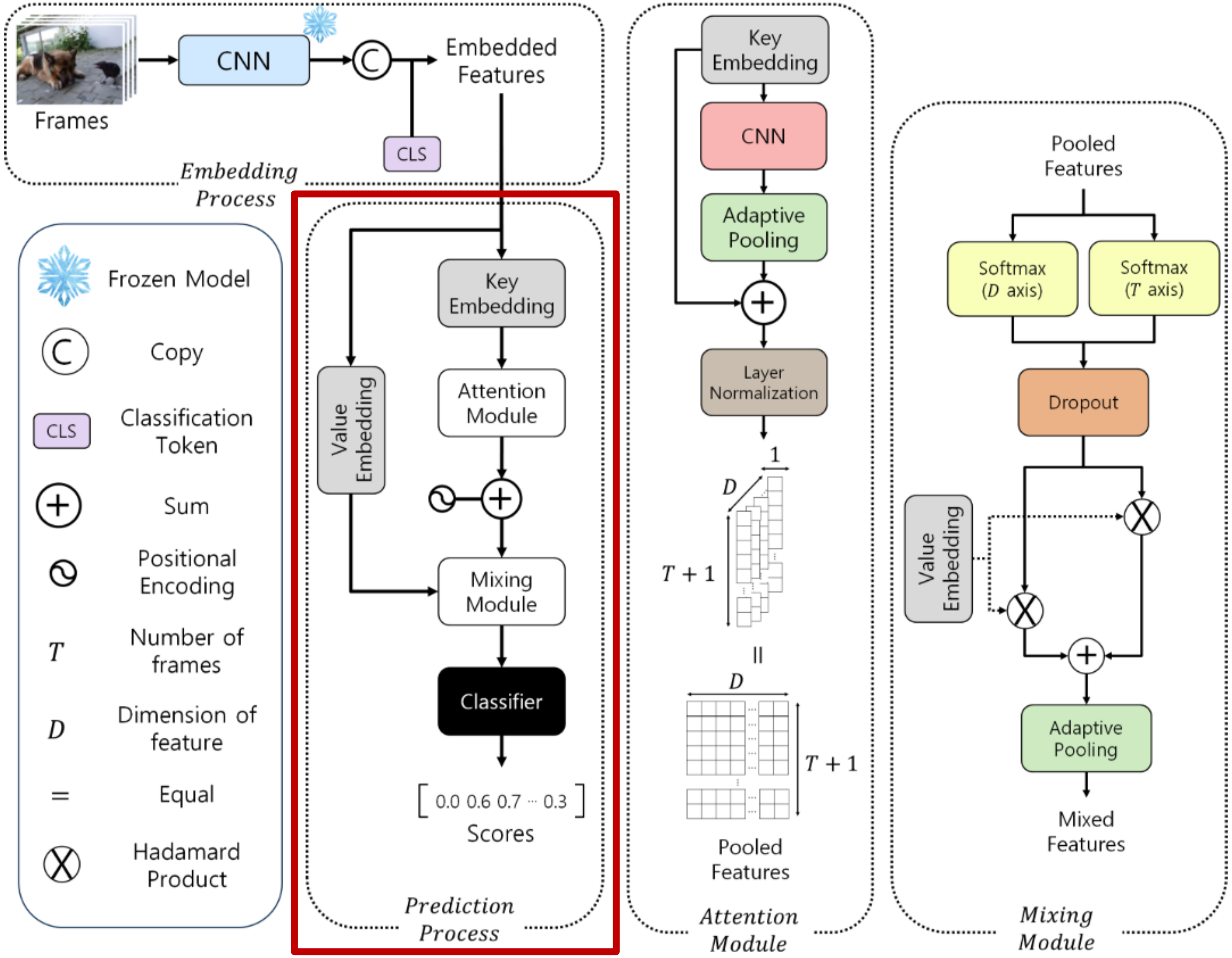

1. Frozen 2D CNN models (GoogleNet) extract features from frames.

2. <u>Copy frame features two times</u> to utilize CNN models better, which are tailored for RGB images. (3 channels)

3. Generate Embedded Features by <u>concatenating the CLS token with frame features</u>, motivated by STVT(2023, ITIP).

# Architecture: Prediction Process

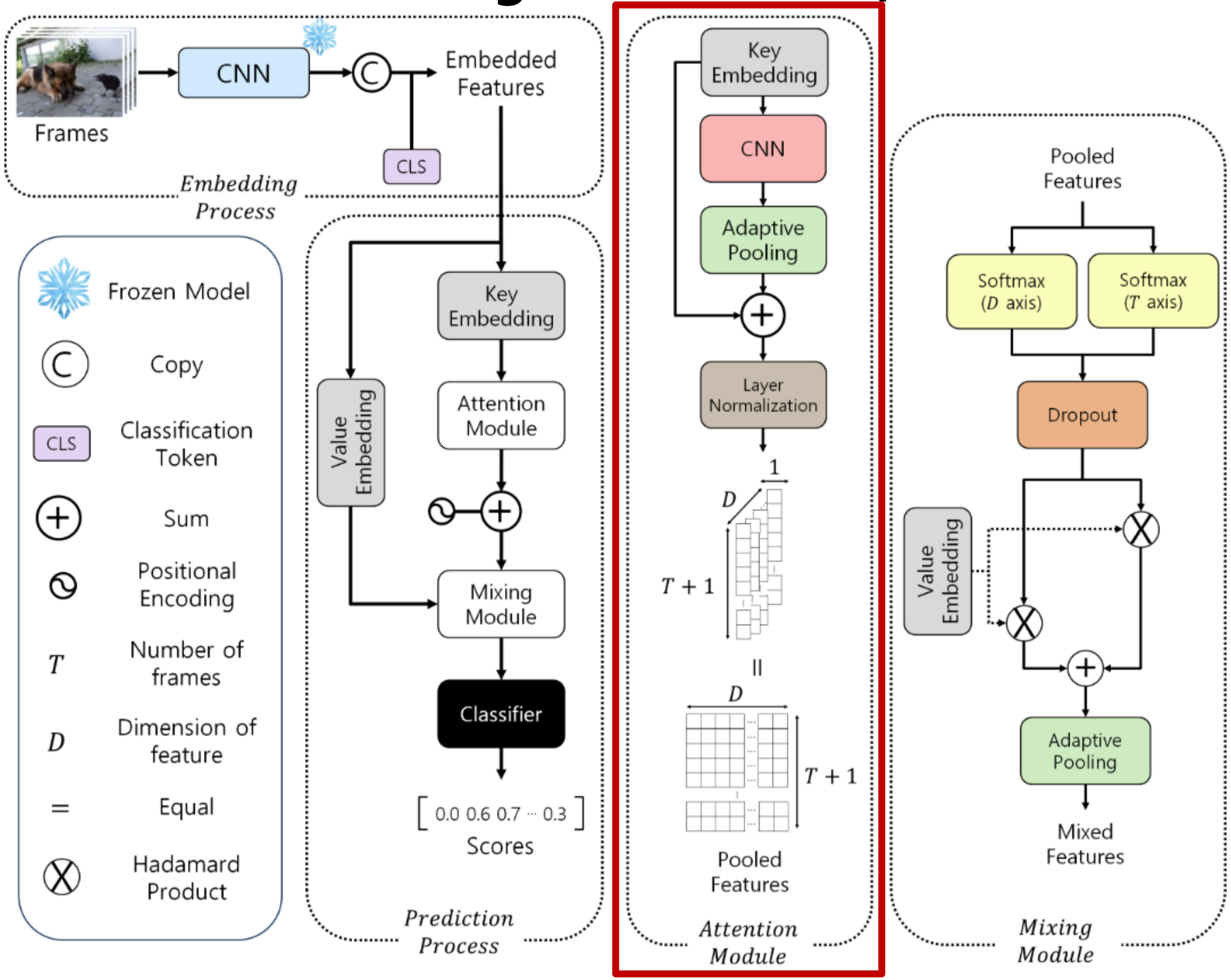
## Predict scores using mixtures of Embedded Features and Pooled Features




1. Generate key and value from Embedded Features using key and value embedding.
2. Attention Module takes the key and creates Pooled Features (weights of Embedded Features).
3. Add positional encodings into Pooled Features.
4. Mix Pooled Features with value by Mixing Module, and input it into the classifier to predict scores.

# Architecture: Attention Module

## Produce the weighted values (Pooled Features) by using 2D CNN models



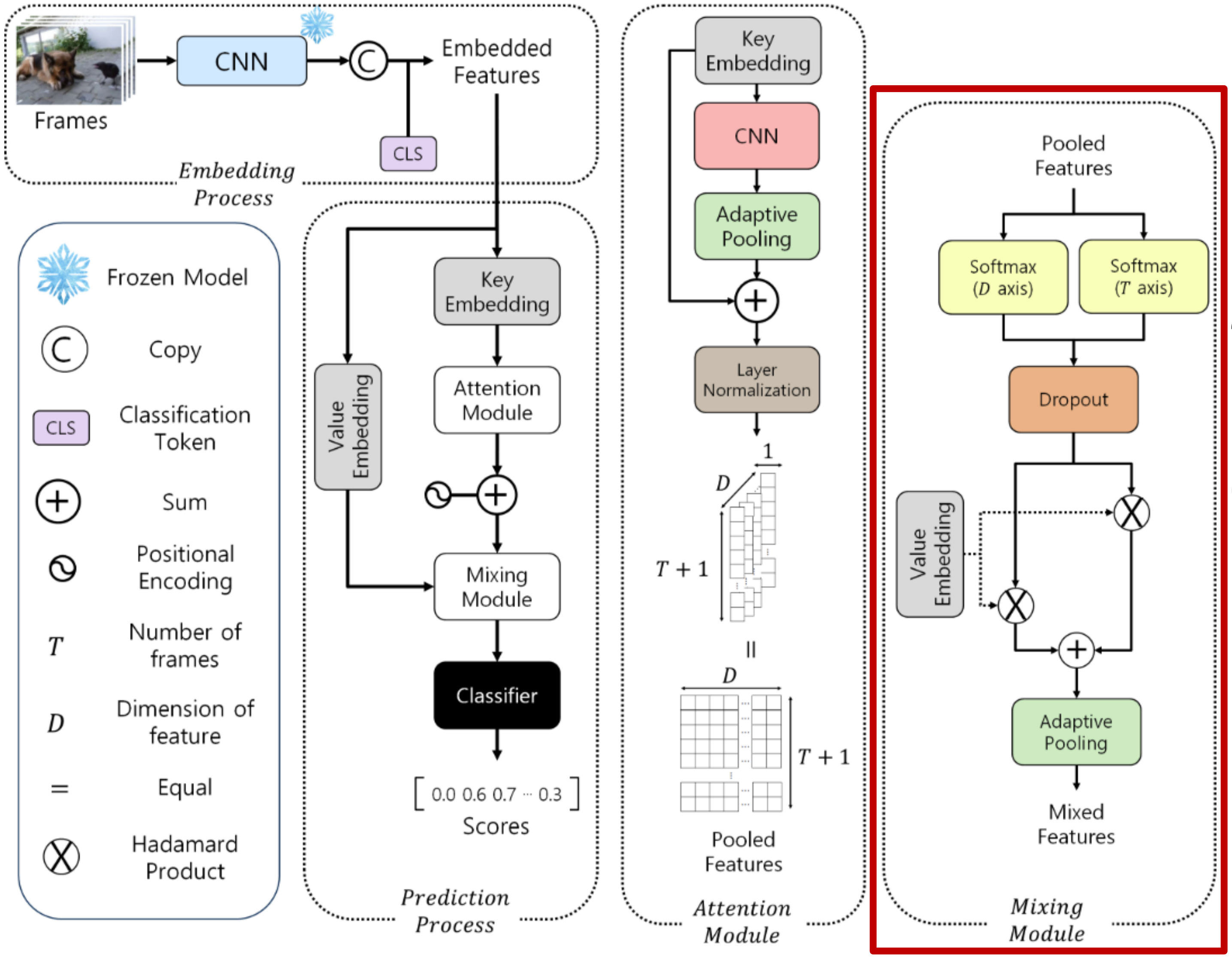

1. <u>Input the key into trainable 2D CNN models</u> (GoogleNet) to reflect key attributes.

2. <u>Employ adaptive pooling</u> to match shape of weights with inputs, and mix that weights with those inputs.

3. For better training, use skip connection and layer normalization, and make Pooled Features.

# Architecture: Mixing Module

**Make attention maps from Pooled Features, and mix them with value**



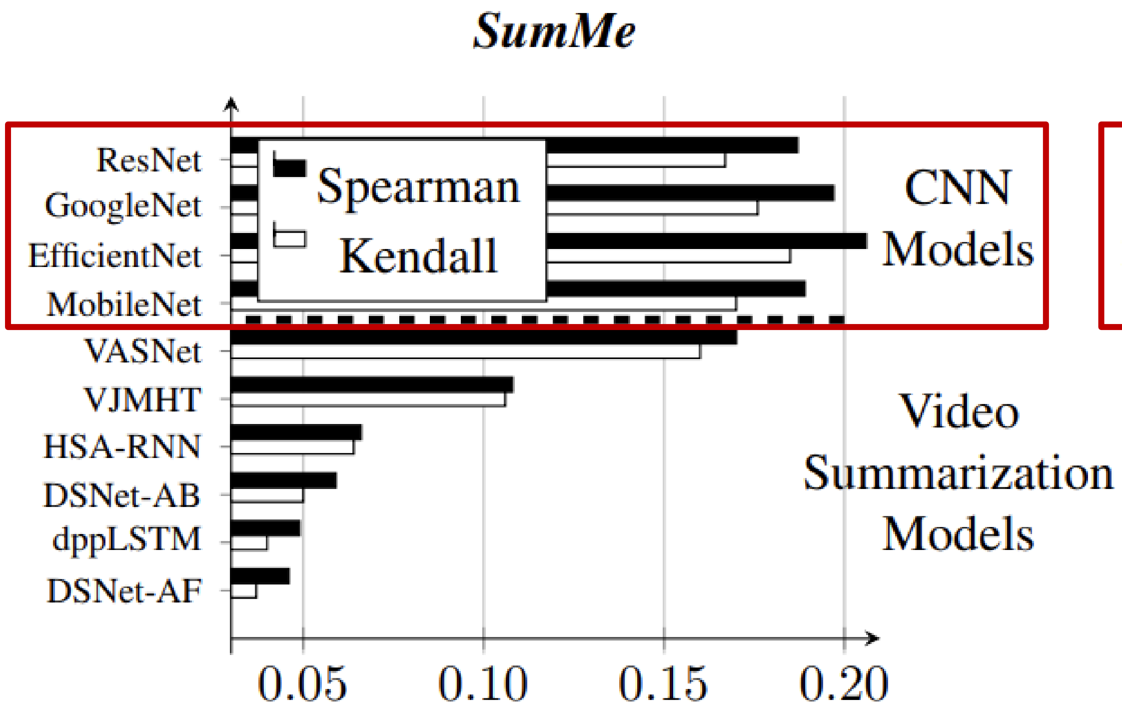

1. <u>Apply softmax along the time and dimension axis,</u> and make spatiotemporal attention maps.

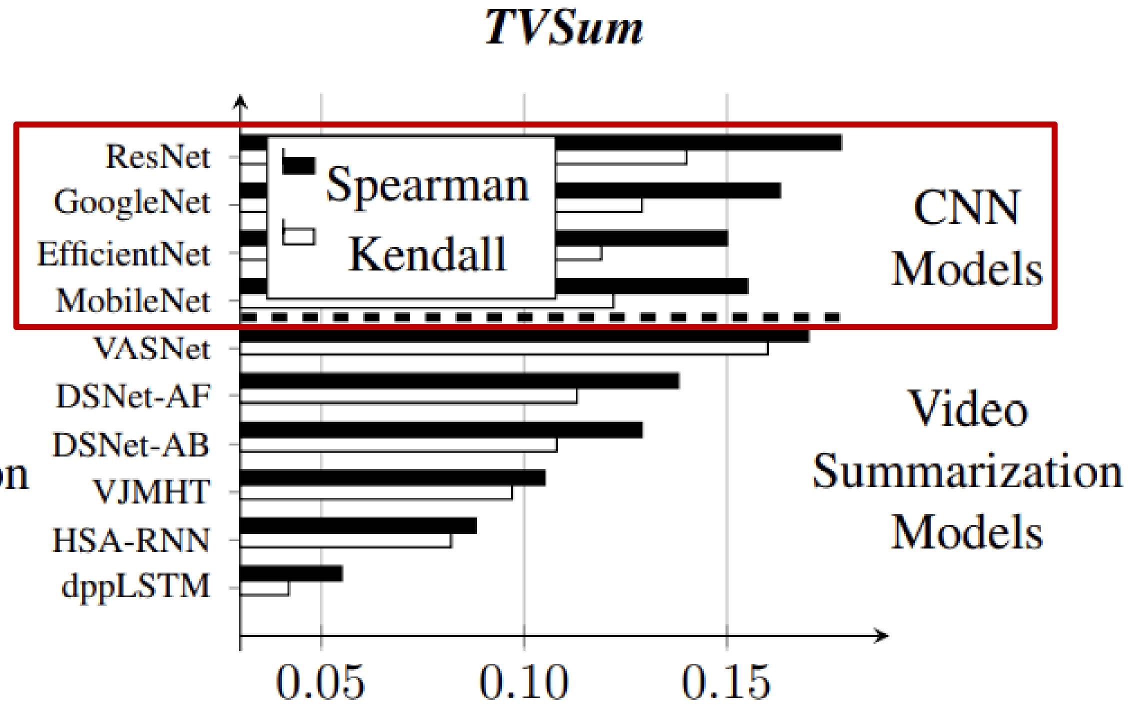
2. After using dropout, mix attention maps with value.

3. <u>Utilize adaptive pooling</u> to adjust the size of outputs.

# Experiment: Prove CNN as the attention mechanism

**CNN can work as the attention based on video summarization results**




1. Target score of each frame ranges from 0 to 1, <u>the same as the weighted values</u>.
2. Good video summarization performance indicates <u>the models are good as the attention algorithm</u>.
3. CNN shows better summarization results than previous video summarization models, meaning it can be used as the attention mechanism.

# Experiment: Performance comparison

## CSTA achieves state-of-the-art results based on the overall performance

| Method | SumMe | | | TVSum | | |
|---|---|---|---|---|---|---|
| | Rank | $\tau$ | $\rho$ | Rank | $\tau$ | $\rho$ |
| Random | - | 0.000 | 0.000 | - | 0.000 | 0.000 |
| Human | - | 0.205 | 0.213 | - | 0.177 | 0.204 |
| dppLSTM[42] | 15 | 0.040 | 0.049 | 22 | 0.042 | 0.055 |
| DAC[8]$^{T}$ | 12.5 | 0.063 | 0.059 | 21 | 0.058 | 0.065 |
| HSA-RNN[45] | 11.5 | 0.064 | 0.066 | 19.5 | 0.082 | 0.088 |
| DAN[27]$^{ST}$ | - | - | - | 19.5 | 0.071 | 0.099 |
| STVT[15]$^{ST}$ | - | - | - | 15.5 | 0.100 | 0.131 |
| DSNet-AF[47]$^{T}$ | 16 | 0.037 | 0.046 | 13.5 | 0.113 | 0.138 |
| DSNet-AB[47]$^{T}$ | 13.5 | 0.051 | 0.059 | 15 | 0.108 | 0.129 |
| HMT[46]$^{M}$ | 10.5 | 0.079 | 0.080 | 17.5 | 0.096 | 0.107 |
| VJMHT[24]$^{T}$ | 8.5 | 0.106 | 0.108 | 17.5 | 0.097 | 0.105 |
| CLIP-It[29]$^{M}$ | - | - | - | 13.5 | 0.108 | 0.147 |

| Method | SumMe | | | TVSum | | |
|---|---|---|---|---|---|---|
| | Rank | $\tau$ | $\rho$ | Rank | $\tau$ | $\rho$ |
| iPTNet[19]$^{+}$ | 8.5 | 0.101 | 0.119 | 11 | 0.134 | 0.163 |
| A2Summ[13]$^{M}$ | 7 | 0.108 | 0.129 | 10 | 0.137 | 0.165 |
| VASNet[7]$^{T}$ | 6 | 0.160 | 0.170 | 9 | 0.160 | 0.170 |
| AAAM[37]$^{T}$ | - | - | - | 6.5 | 0.169 | 0.223 |
| MAAM[37]$^{T}$ | - | - | - | 5.5 | 0.179 | 0.236 |
| VSS-Net[43]$^{ST}$ | - | - | - | 3 | 0.190 | 0.249 |
| DMASum[39]$^{ST}$ | 11 | 0.063 | 0.089 | **1** | **0.203** | **0.267** |
| RR-STG[48]$^{ST}$ | 2.5 | 0.211* | 0.234 | 7.5 | 0.162 | 0.212 |
| MSVA[9]$^{M}$ | 3.5 | 0.200 | 0.230 | 5.5 | 0.190 | 0.210 |
| SSPVS[25]$^{M}$ | 3* | 0.192 | 0.257* | 4.5 | 0.181 | 0.238 |
| GoogleNet[35]$^{ST}$ | 5 | 0.176 | 0.197 | 11.5 | 0.129 | 0.163 |
| CSTA$^{ST}$ | **1** | **0.246** | **0.274** | 2* | 0.194* | 0.255* |

*Rank: Average performance rank between SumMe and TVSum datasets, τ: Kendall's coefficients, ρ: Spearman's coefficients*

1. Considering the average rank for both SumMe and TVSum datasets, CSTA shows the best results.
2. DMASum performs slightly better than CSTA on TVSum, but much poorer on SumMe.
3. CSTA outperforms other spatiotemporal attention-based models thanks to the CNN.

# Experiment: Computation analysis

## CSTA shows better trade-offs between results and efficiency than others

| Method | SumMe | | | TVSum | | |
|---|---|---|---|---|---|---|
| | Rank | *FE* | *SP* | Rank | *FE* | *SP* |
| DSNet-AF[47] $^{T}$ | 16 | 413.03G | 1.18G | 13.5 | 661.83G | 1.90G |
| DSNet-AB[47] $^{T}$ | 13.5 | 413.03G | 1.29G | 15 | 661.83G | 2.07G |
| VJMHT[24] $^{T}$ | 8.5 | 413.03G | 18.21G | 17.5 | 661.83G | 28.25G |
| VASNet[7] $^{T}$ | 6 | 413.03G | 1.43G | 9 | 661.83G | 2.30G |
| RR-STG[48] $^{ST}$ | 2.5 | 54.82T | 0.31G | 7.5 | 88.41T | 0.20G |
| MSVA[9] $^{M}$ | 3.5 | 13.76T | 3.63G | 5.5 | 22.08T | 5.81G |
| SSPVS[25] $^{M}$ | 3 | 413.49G | 20.72G | 4.5 | 662.46G | 44.22G |
| CSTA $^{ST}$ | 1 | 413.03G | 9.78G | 2 | 661.83G | 15.73G |

*Rank: Average performance rank between SumMe and TVSum datasets*
*FE: MACs for Feature Extraction, SP: MACs for Score Prediction*

1. Measure MACs of feature extractions (FE) and score predictions (SC).
2. Based on the average rank of performance,
   good summarization scores require huge costs or multi-modal data.
3. CSTA demands fewer MACs with the best performance.

# Conclusion

## Summary

-In video summarization, considering temporal and spatial attention is necessary.
-Embracing spatiotemporal attention requires huge resources for better results.
-For efficient way, we propose CSTA relying on CNN,
having position awareness and efficiency.

## Contribution

1. This is the first paper to apply 2D CNN to frame features in video summarization.
2. Propose CSTA as the efficient spatiotemporal attention algorithm.
3. CSTA shows the best results based on the overall performance.

# Thank You!

## For more details,

**Poster 20/06/2024 THU 17:00 - 19:00**

**Paper:**

**https://arxiv.org/abs/2405.11905**

**Code:**

**https://github.com/thswodnjs3/CSTA**