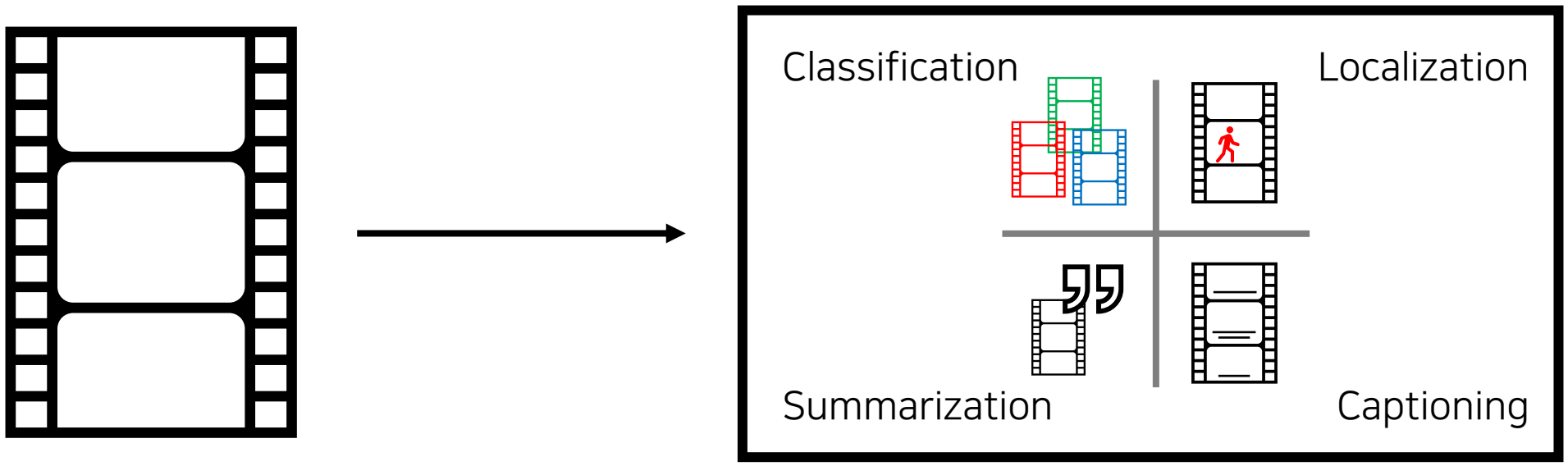
CVPR |  ECV24

LVS: A Learned Video Storage for Fast and Efficient Video Understanding

Yunghee Lee, Jongse Park
KAIST

CASYS | KAIST Computer Architecture & System Lab

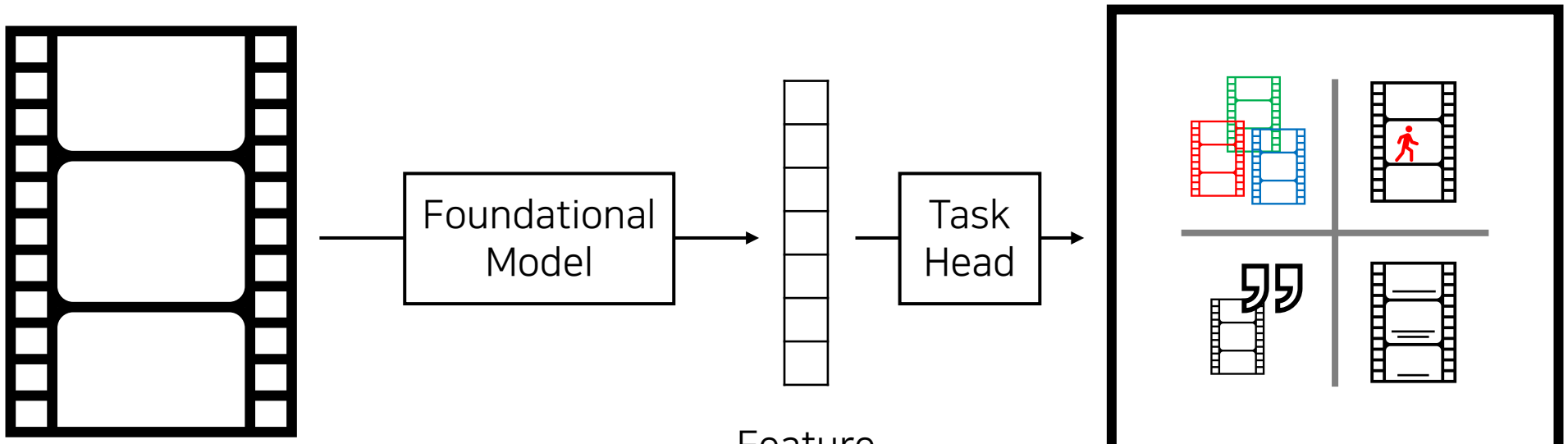
Video Understanding (VU)



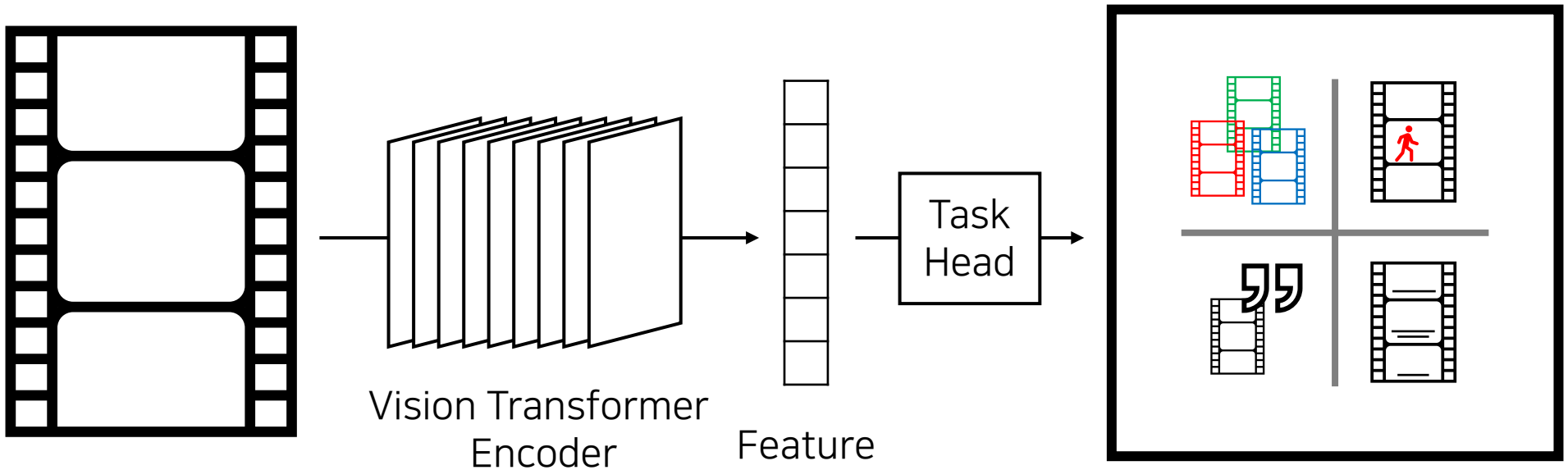
Video data

High-level insight

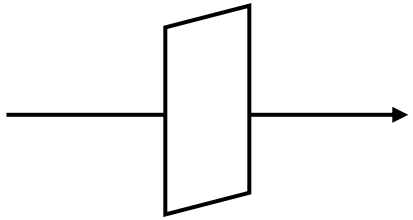
Foundational Model (FM)



FMs Use Transformers

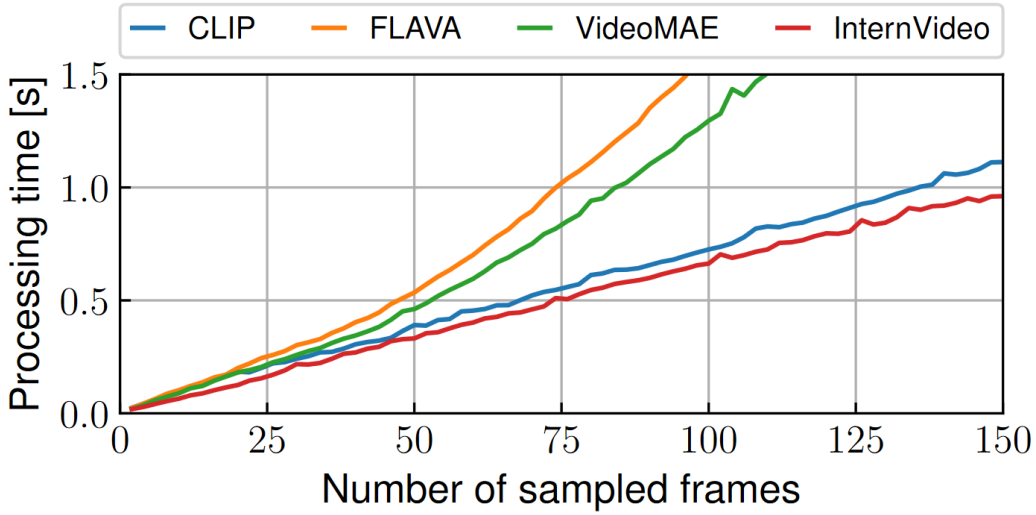


Computational Cost of Transformers



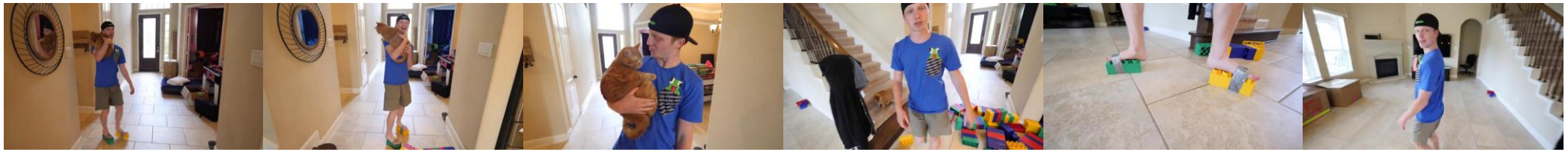
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_\text{K}}}\right)\mathbf{V}$$

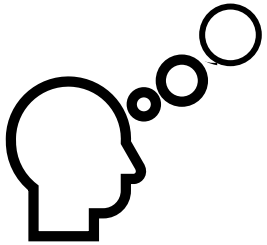
Quadratic complexity w.r.t. sequence length



VU over long videos (which means long sequence length) are impractical

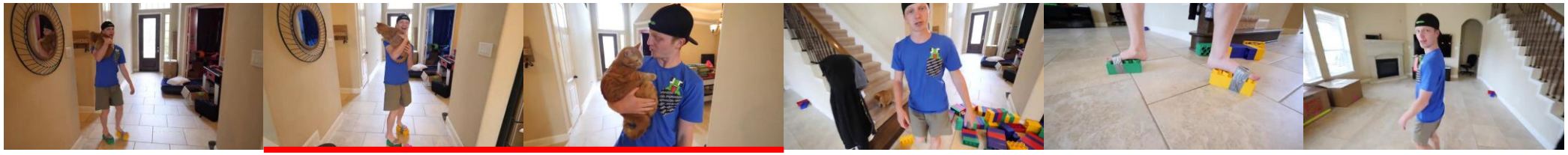
How Humans Understand Long Video





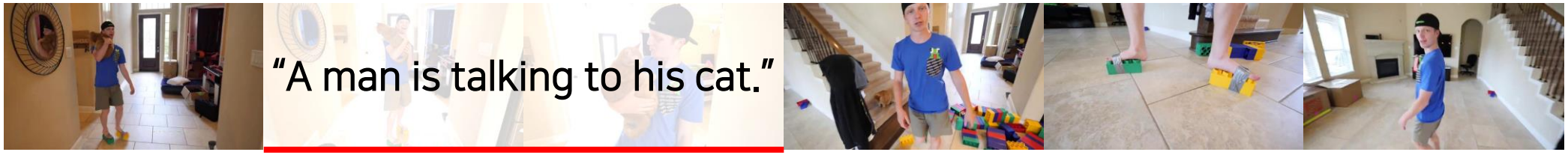
Let’s watch and summarize the clip…

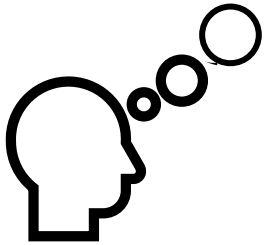
How Humans Understand Long Video



"A man is talking to his cat."

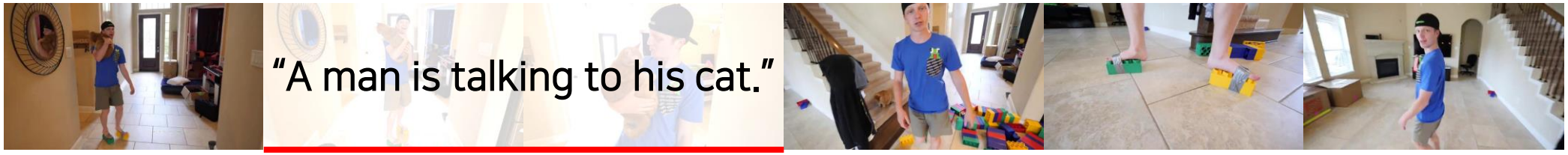
How Humans Understand Long Video

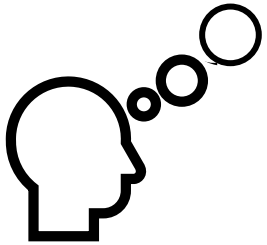




Now let’s summarize the whole video…

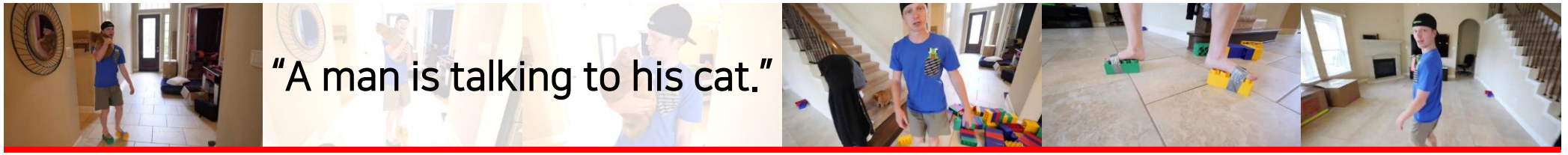
How Humans Understand Long Video





Now let’s summarize the whole video… But should we watch again?

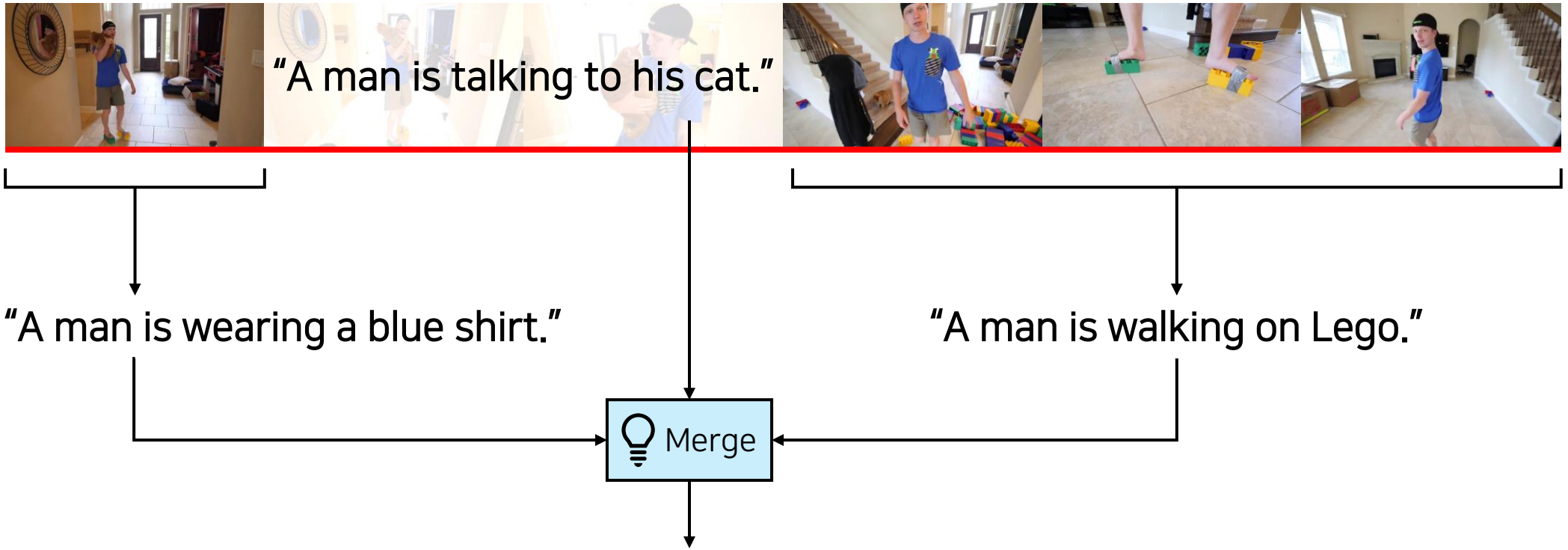
How Humans Understand Long Video



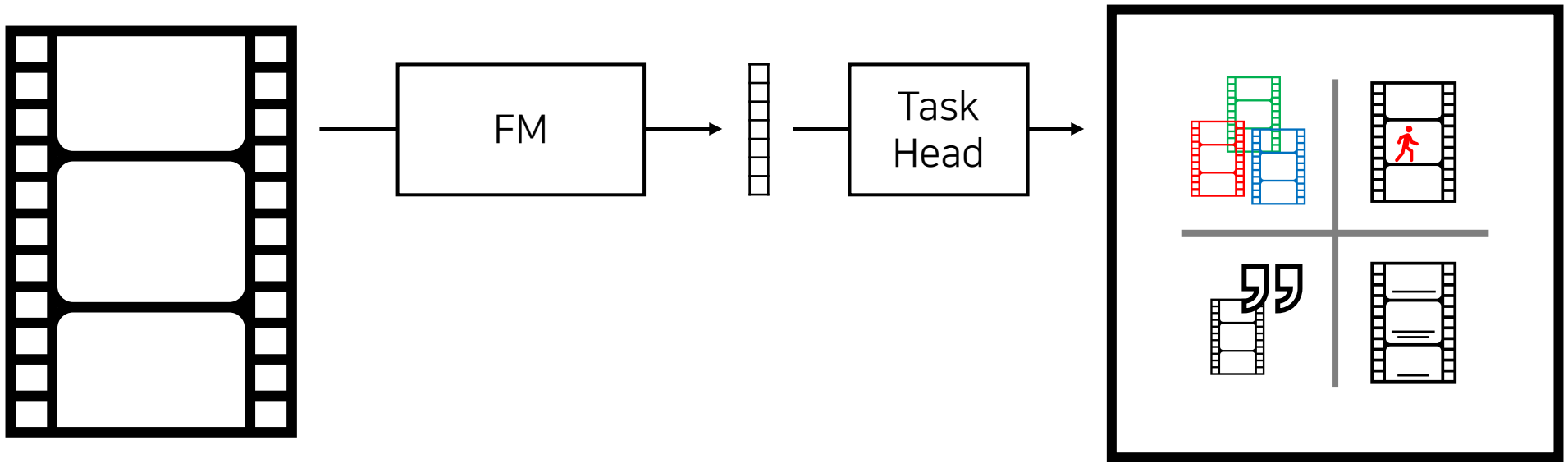
"A man is wearing a blue shirt."

"A man is walking on Lego."

How Humans Understand Long Video



LVS: Learned Video Storage

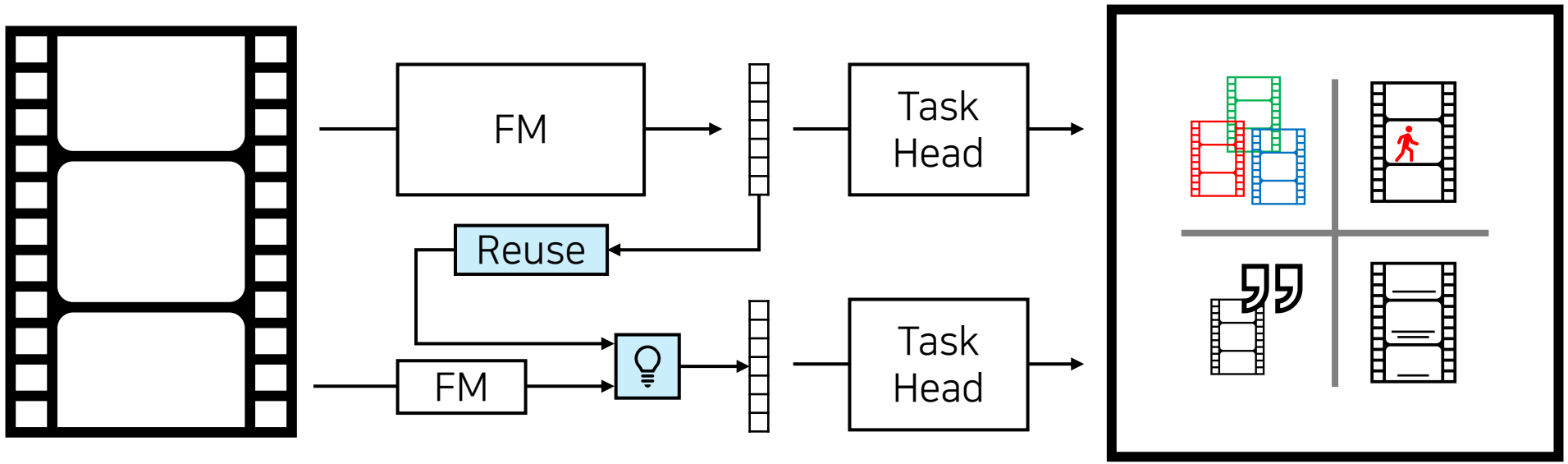


Video data

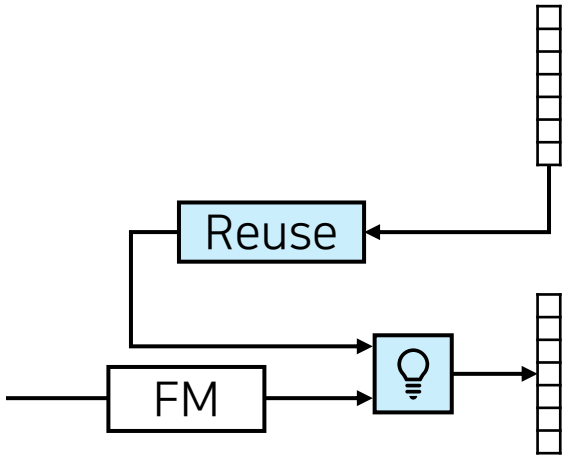
Feature

High-level insight

LVS: Learned Video Storage



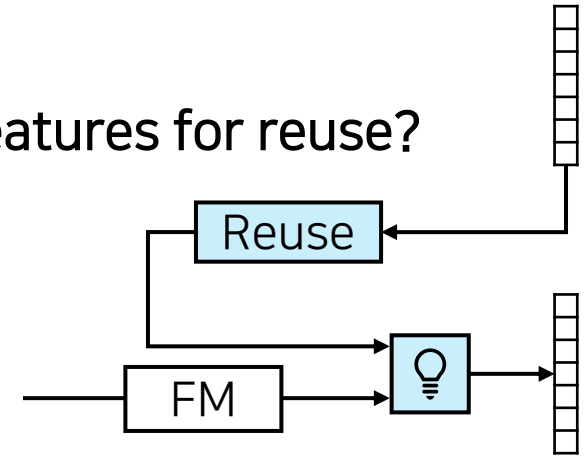
Challenges



1. How to implement merge operation?

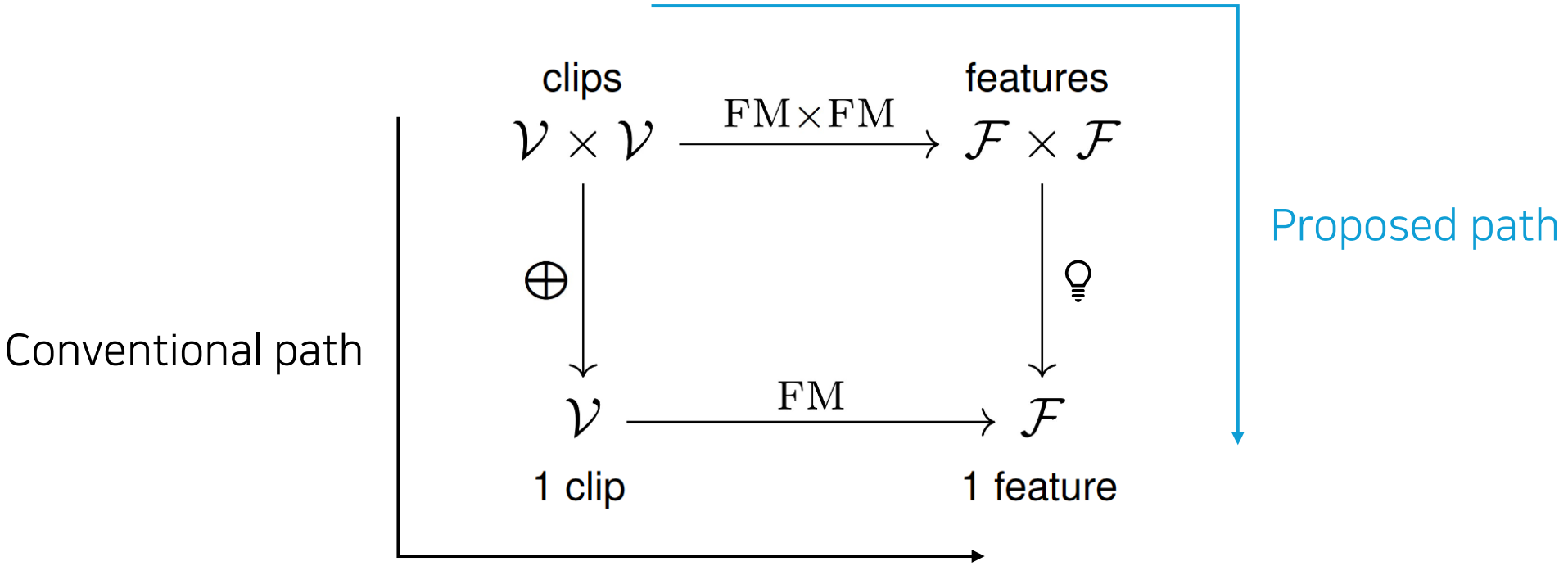
Challenges

2. How to select features for reuse?



1. How to implement merge operation?

Requirement

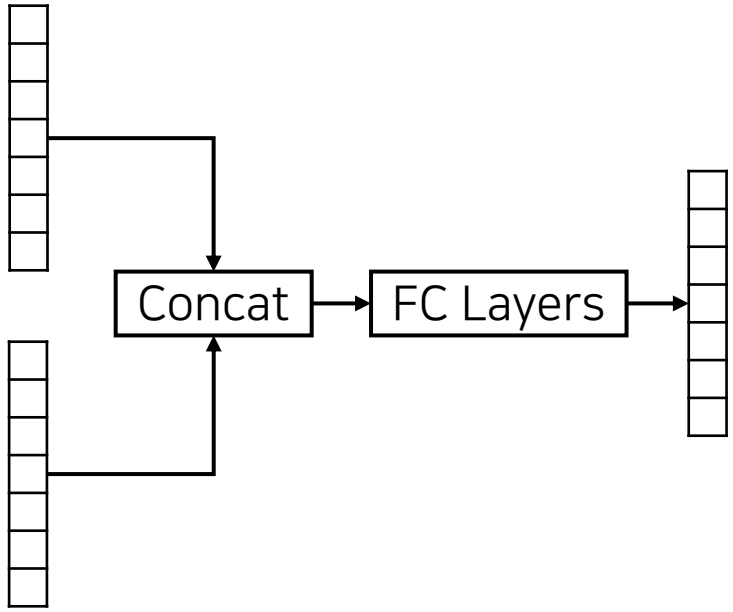


Computing feature from subfeatures should give same results with using full video

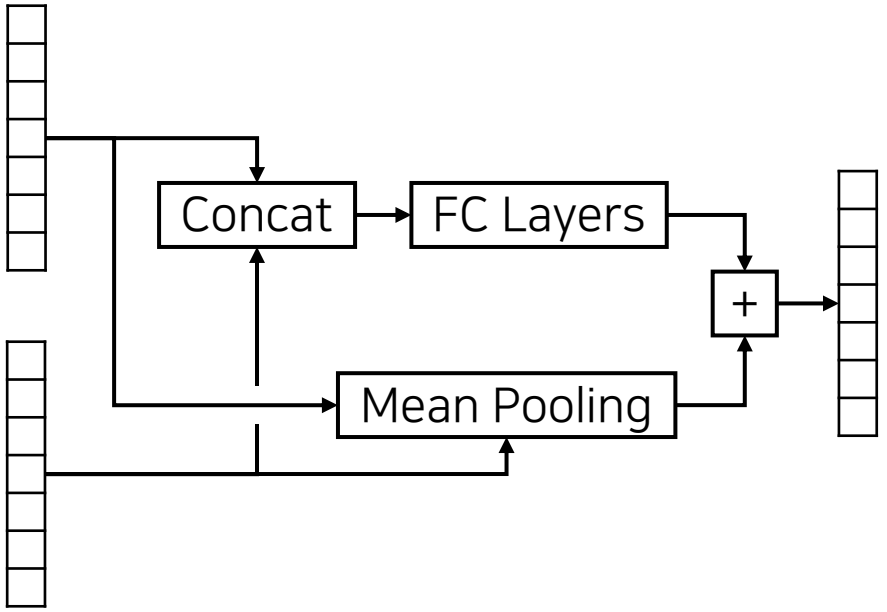
(or, 💡 should be a monoid homomorphism)

MLP

Such feature fusion can be approximated with MLP

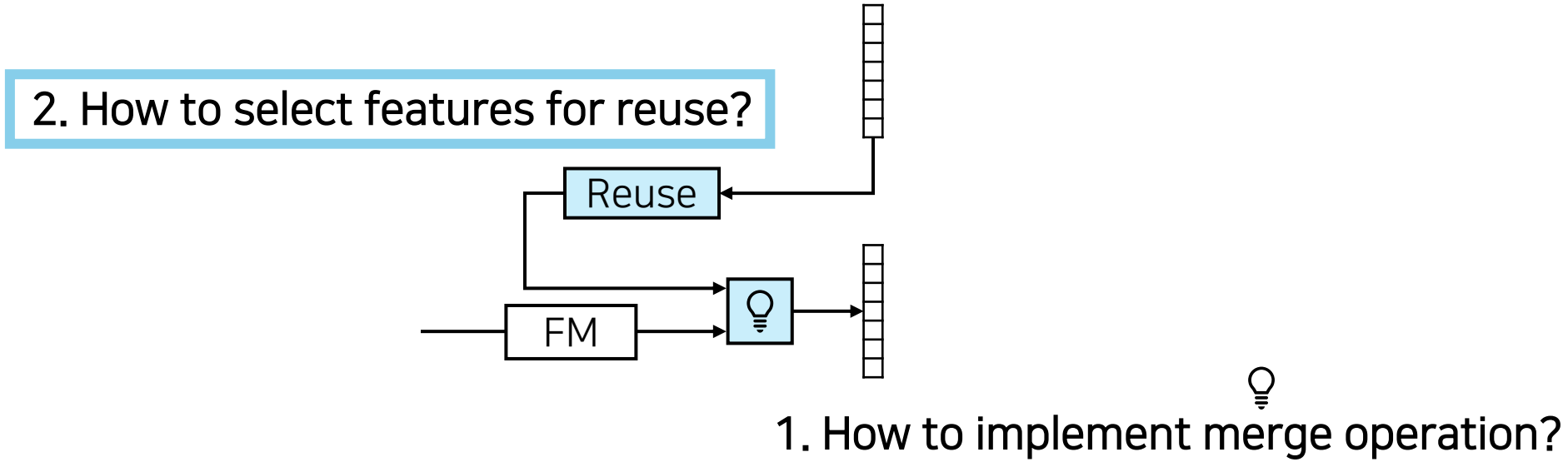


A) MLP

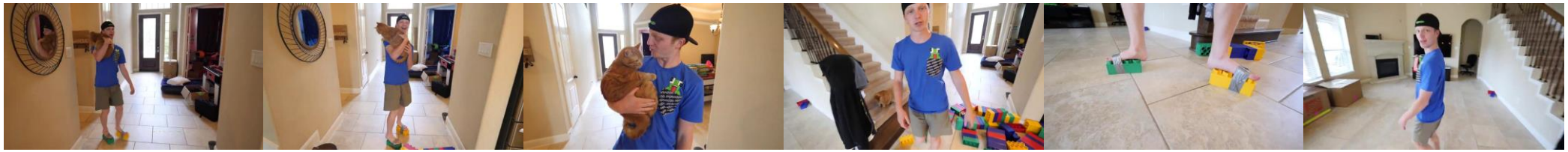


B) MLP + AVG

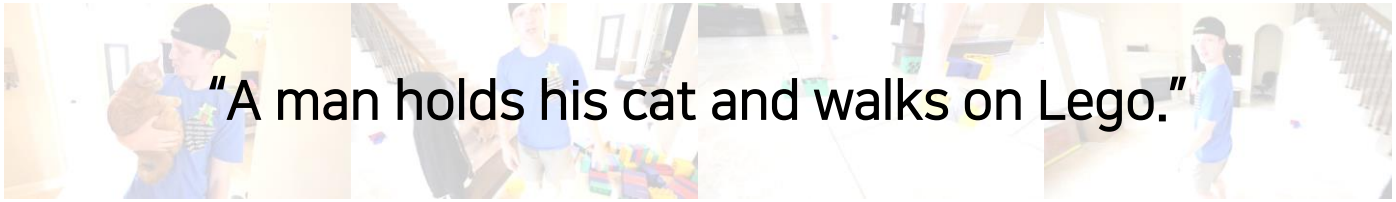
Challenges


2. How to select features for reuse?
Reuse
FM
1. How to implement merge operation?

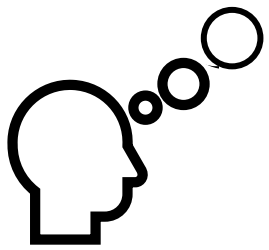
Cost Estimation



"A man is talking to his cat."

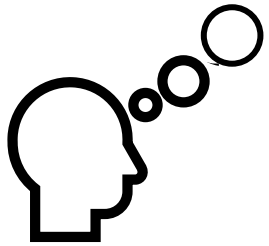


"A man holds his cat and walks on Lego."



Which memoized feature should I pick to summarize the whole video?

Cost Estimation



Let's estimate the cost of FM for each case!

$$\text{COST}(\{c_1, c_2, \cdots, c_n\}) = rn + \sum_{i=1}^{n} m_i l_i$$

where

c_i is the ith subclip being used

$$m_i = \begin{cases} 1, & \text{if } c_i \text{ needs decoding and FM} \\ 0, & \text{if } c_i \text{ is already saved as feature} \end{cases}$$

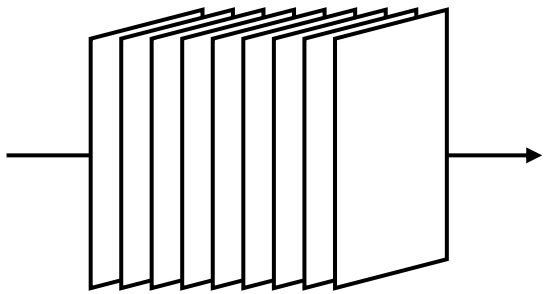
r is a constant factor

l_i is the length of c_i

The case with the least FM cost is selected (using a SMT solver)

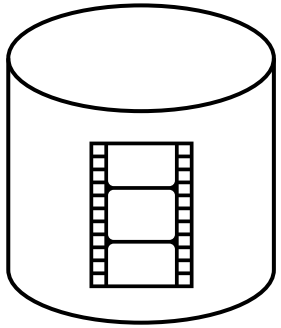
Evaluation: Methodology

Foundational Model



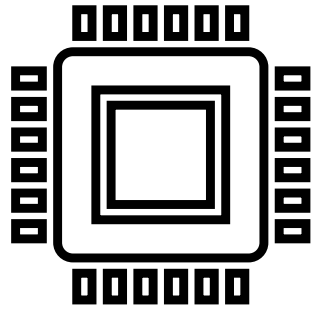
CLIP, FLAVA,
VideoMAE, InternVideo

Dataset



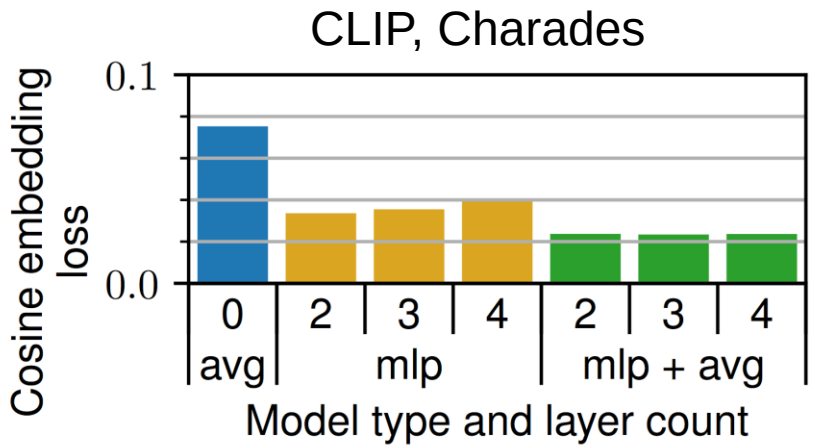
UCF101, MSR-VTT,
Charades, Long Videos

System



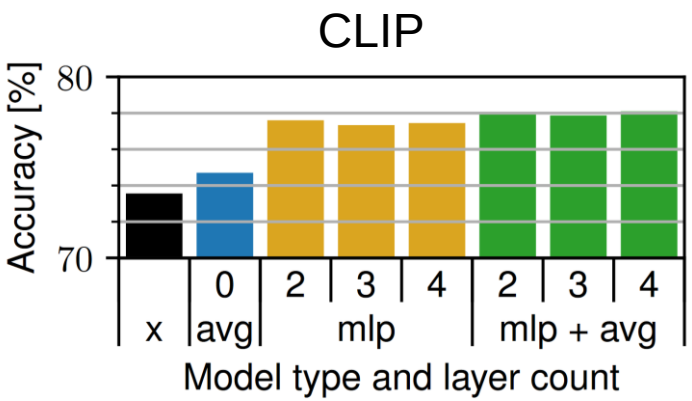
2× Intel Xeon 6326 Gold,
NVIDIA Geforce RTX 3090

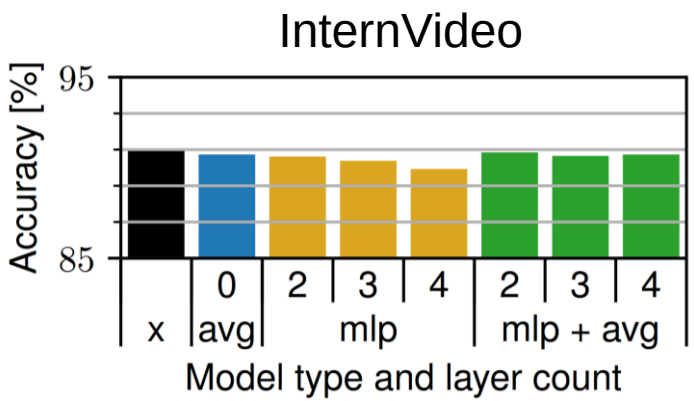
Evaluation: Embedding Accuracy



MLP + Average models give cosine embedding loss <0.025

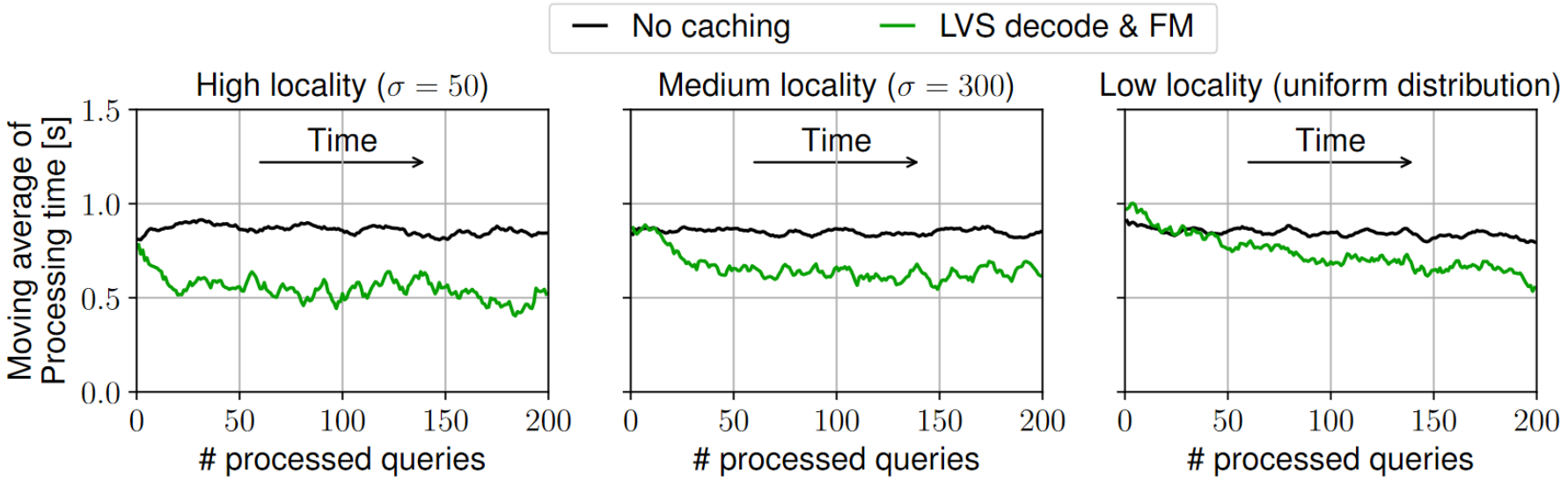
Evaluation: End Task Accuracy





<1% accuracy degradation for UCF101 classification task

Evaluation: Inference Latency



When query locality is high, speedup increases. Up to 1.59× less latency

(Overhead from SMT solver ignored)

Conclusion

- We propose Learned Video Storage (LVS) that reuses features in future queries
- LVS includes MLP-based technique to perform feature fusion
  and feature selection technique using a cost estimation function
- LVS brings up to 1.59× speedup w/ negligible accuracy degradation (<1%)

CVPR |  ECV24

LVS: A Learned Video Storage for Fast and Efficient Video Understanding

Yunghee Lee, Jongse Park

{yhlee, jspark}@casys.kaist.ac.kr

CASYS | KAIST Computer Architecture & System Lab