data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language （META）

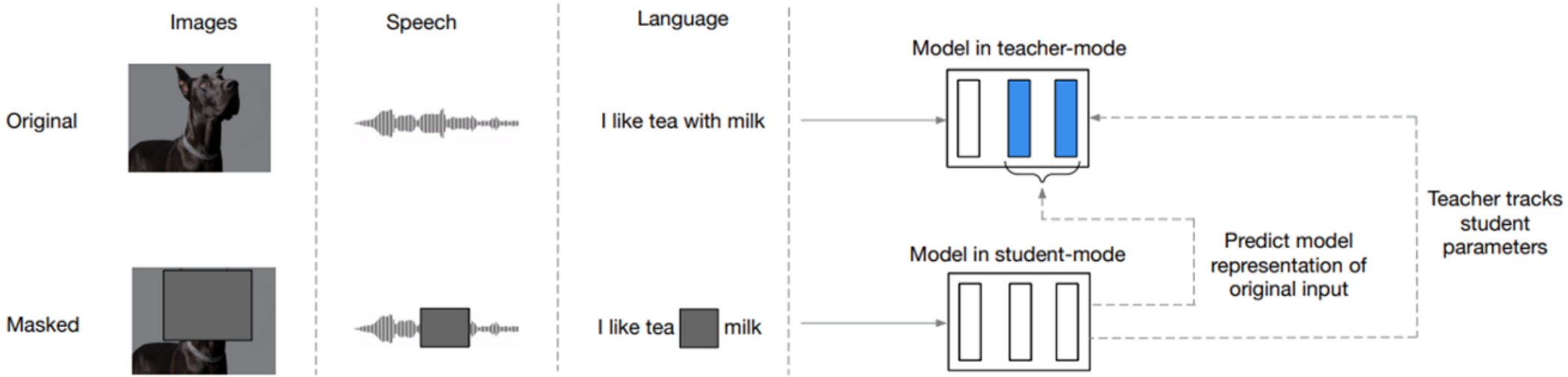
Motivation

- There are currently big differences in the way self-supervised learning algorithms learn from images, speech, text, and other modalities.

  data2vec, the first high-performance self-supervised algorithm that works for multiple modalities ✅

Method Overview

The teacher mode generates representation from a given sample (i.e. image, speech, text). A masked version of the same sample is passed to the student mode. Learning happens by minimizing the objective function between the student's prediction of a target that is constructed by teachers' parameters.



- The target is then constructed by using the top K (closer to the output) blocks of the transformer.

$$y_t = \frac{1}{K} \sum_{l=L-K+1}^{L} \hat{a}_t^l$$

- Objective Function

$$\mathcal{L}(y_t, f_t(x)) = \begin{cases} \frac{1}{2}(y_t - f_t(x))^2/\beta & |y_t - f_t(x)| \leq \beta \\ (|y_t - f_t(x)| - \frac{1}{2}\beta) & \text{otherwise} \end{cases}$$

Results

1. Image (metric: accuracy. Higher value, better performance):

Table 1. Computer vision: top-1 validation accuracy on ImageNet-1K with ViT-B (86M parameters) and ViT-L (307M parameters) models. Our results are based on training for 800 epochs while as several other well-performing models were trained for 1,600 epochs (MAE, MaskFeat).

	ViT-B	ViT-L
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
BEiT (Bao et al., 2021)	83.2	85.2
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.2

2. Speech (metric: word error rate. Lower value, better performance)

Table 2. Speech processing: word error rate on the Librispeech test-other test set when fine-tuning pre-trained models on the Libri-light low-resource labeled data setups (Kahn et al., 2020) of 10 min, 1 hour, 10 hours, the clean 100h subset of Librispeech and the full 960h of Librispeech. Models use the 960 hours of audio from Librispeech (LS-960) as unlabeled data. We indicate the language model used during decoding (LM). Results for all dev/test sets and other LMs can be found in the supplementary material (Table 5).

	Unlabeled data	LM	Amount of labeled data				
			10m	1h	10h	100h	960h
Base models							
wav2vec 2.0 (Baevski et al., 2020b)	LS-960	4-gram	15.6	11.3	9.5	8.0	6.1
HuBERT (Hsu et al., 2021)	LS-960	4-gram	15.3	11.3	9.4	8.1	-
WavLM (Chen et al., 2021a)	LS-960	4-gram	-	10.8	9.2	7.7	-
data2vec	LS-960	4-gram	12.3	9.1	8.1	6.8	5.5

3. Texts (metric: GLUE score. Higher value, better performance)

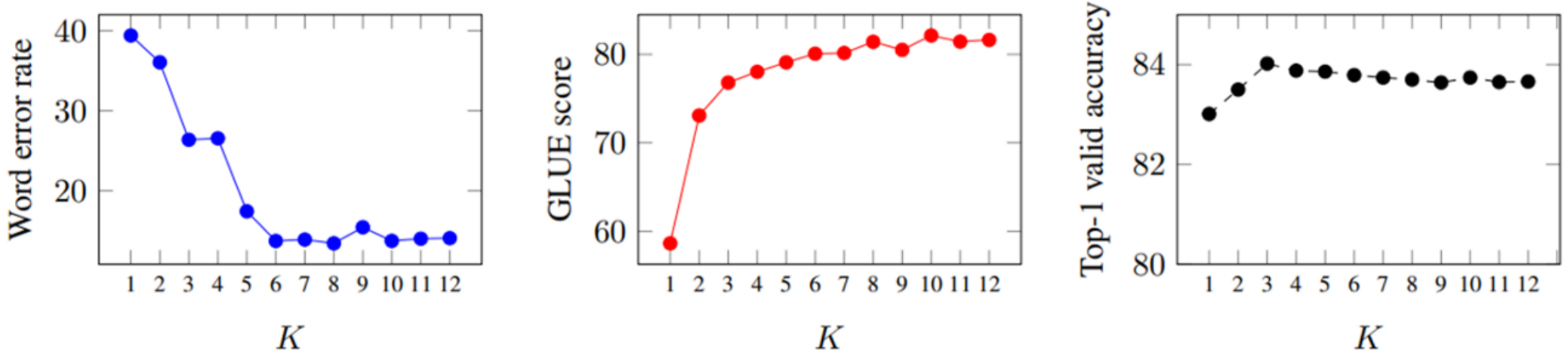
Table 3. Natural language processing: GLUE results on the development set for single-task fine-tuning of individual models. For MNLI we report accuracy on both the matched and unmatched dev sets, for MRPC and QQP, we report the unweighted average of accuracy and F1, for STS-B the unweighted average of Pearson and Spearman correlation, for CoLA we report Matthews correlation and for all other tasks we report accuracy. BERT Base results are from Wu et al. (2020) and our baseline is RoBERTa re-trained in a similar setup as BERT. We also report results with wav2vec 2.0 style masking of spans of four BPE tokens with no unmasked tokens or random targets.

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
Base models									
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9

Ablation Study

1. Top K blocks

The paper argues that using the average of top K blocks in the teacher mode is better than using just the top one



(a) Speech (b) NLP (c) Vision

2. Target Feature Type

Rather than just use the top K blocks, the authors also tried using different parts of the teacher mode and found that using the FFN is the best.

Table 4. Effect of using different features from the teacher model as targets: we compare using the output of the self-attention module, the feed-forward module (FFN) as well as after the final residual connection (FFN + residual) and layer normalization (End of block). We pre-train speech models on Librispeech, fine-tune with 10 hours of labeled data and report WER on dev-other without a language model. Results are not directly comparable to the main results since we train for 200K updates.

Layer	WER
self-attention	100.0
FFN	13.1
FFN + residual	14.8
End of block	14.5

Conclusion

- The paper introduces a new general self-supervised learning framework and achieves SOTA performance for three modalities.

Personal Remarks:

- It’d be more interesting to see how this method performs for unstructured modality, e.g. graphs
- Transformer, as a flexible architecture not constraint to a specific modality, plays an important role in the success of this method.
- This work serves as a key step for unifying inputs from different modalities.

MTAG: Modal-Temporal Attention Graph for Unaligned Human Multimodal Language Sequences

Jianing Yang[1]*, Yongxin Wang[1]*, Ruitao Yi[1], Yuying Zhu[1], Azaan Rehman[1], Amir Zadeh[1], Soujanya Poria[2], Louis-Philippe Morency[1]

[1]Carnegie Mellon University
[2]Singapore University of Technology and Design

{jianing3, yongxinw, ruitaoy, yuyingz, arehman, abagherz}@cs.cmu.edu, sporia@sutd.edu.sg, morency@cs.cmu.edu

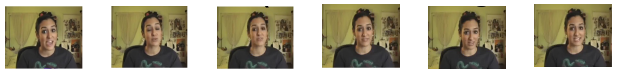
NAACL 2021

What is multimodal sequence?

- Data that is ***sequential*** and recorded in ***multiple*** channels

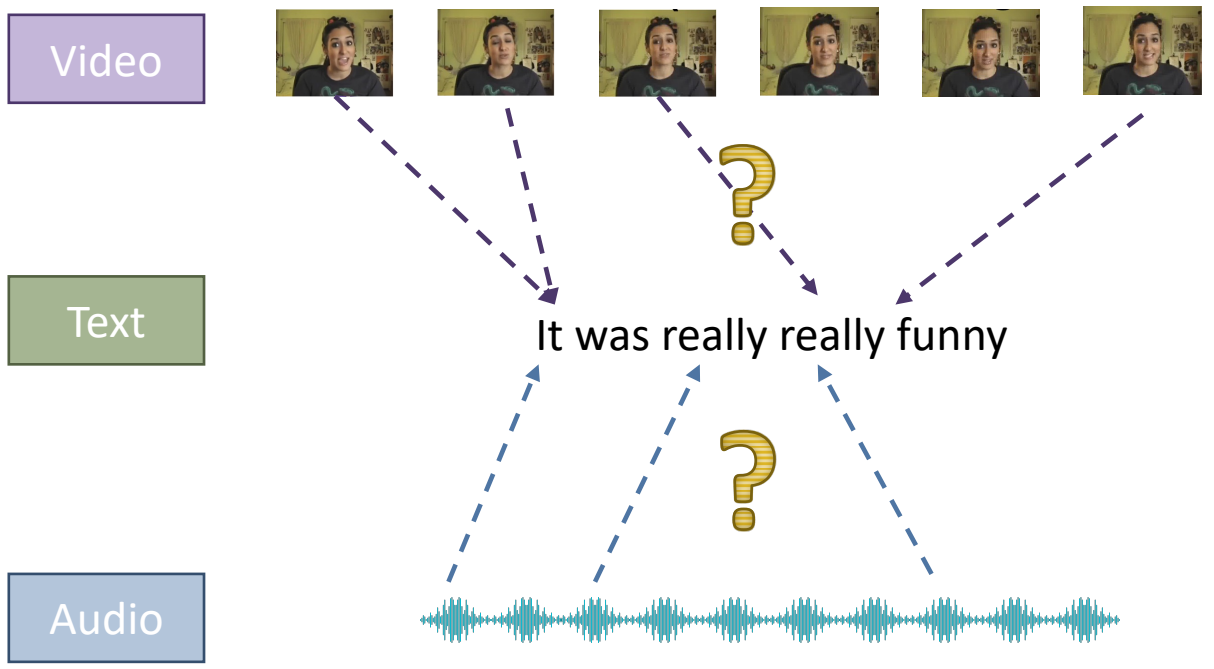
Multimodal Sequence Challenges

- Irregular lengths
- Misalignment
- Fusion of more than 2 modalities
- Long-term temporal dependency

Modality	Feature Length
Video 	6
Text It was really really funny	5
Audio 	10

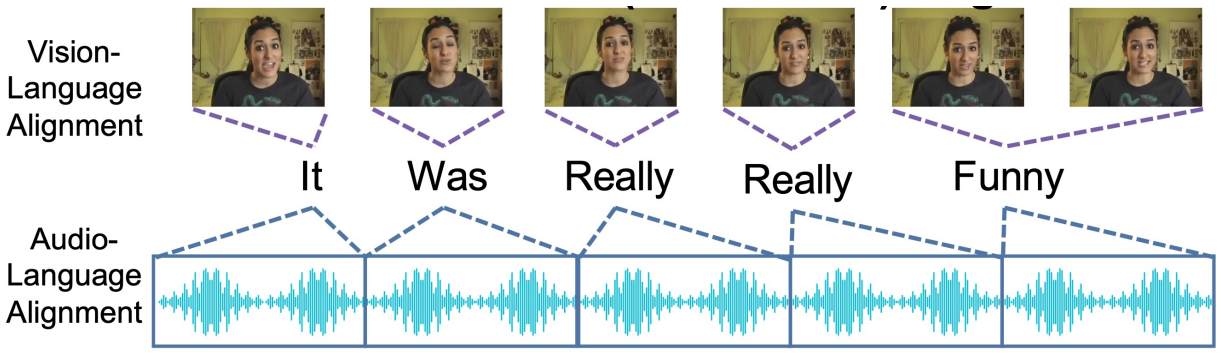
Multimodal Sequence Challenges

- Irregular lengths
- Misalignment
- Fusion of more than 2 modalities
- Long-term temporal dependency



Multimodal Sequence Challenges

- Irregular lengths
- Misalignment
- Fusion of more than 2 modalities
- Long-term temporal dependency



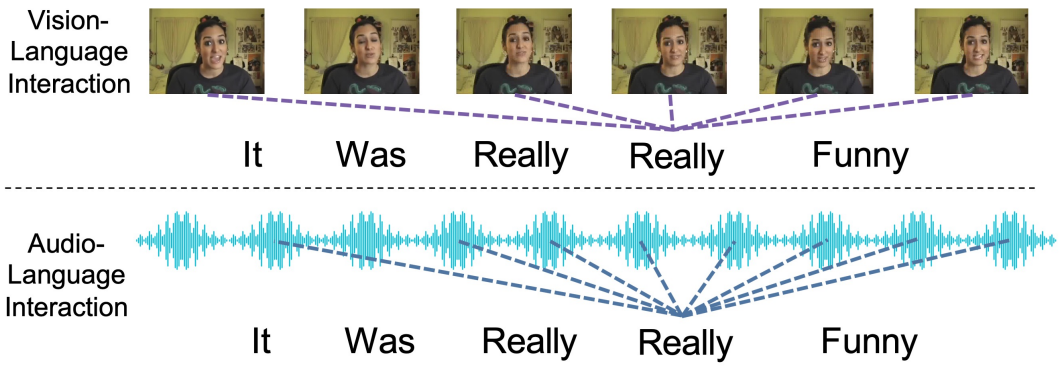
Hard (word-level) Alignment (prior approach)

Cons:

- More supervision
  - (e.g. time intervals)
- More engineering effort

Multimodal Sequence Challenges

- Irregular lengths
- Misalignment
- Fusion of more than 2 modalities
- Long-term temporal dependency



Pairwise Cross-modal Attention (Prior approach)

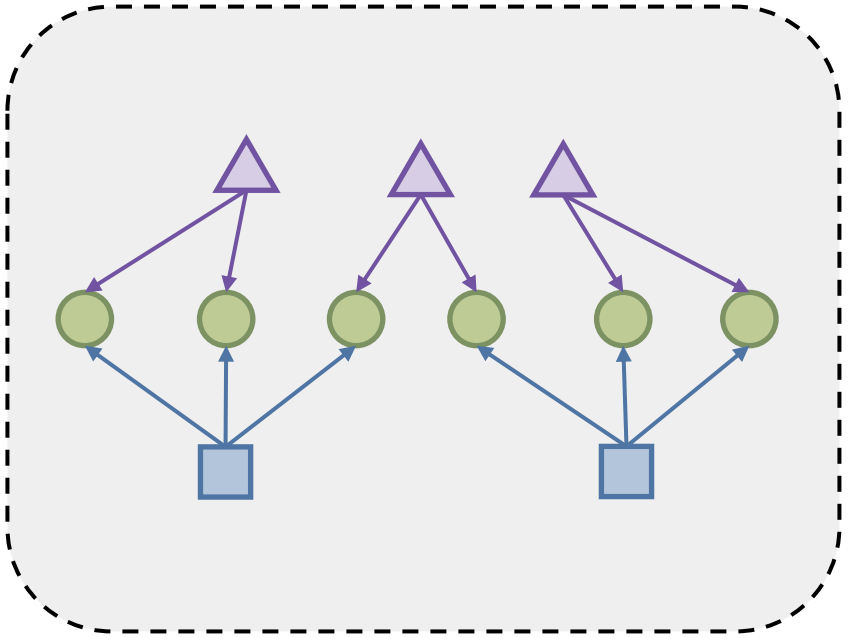
Cons:

- Only two modalities at a time
- Repeated many times for each modality pair
  --> Lots of model parameters

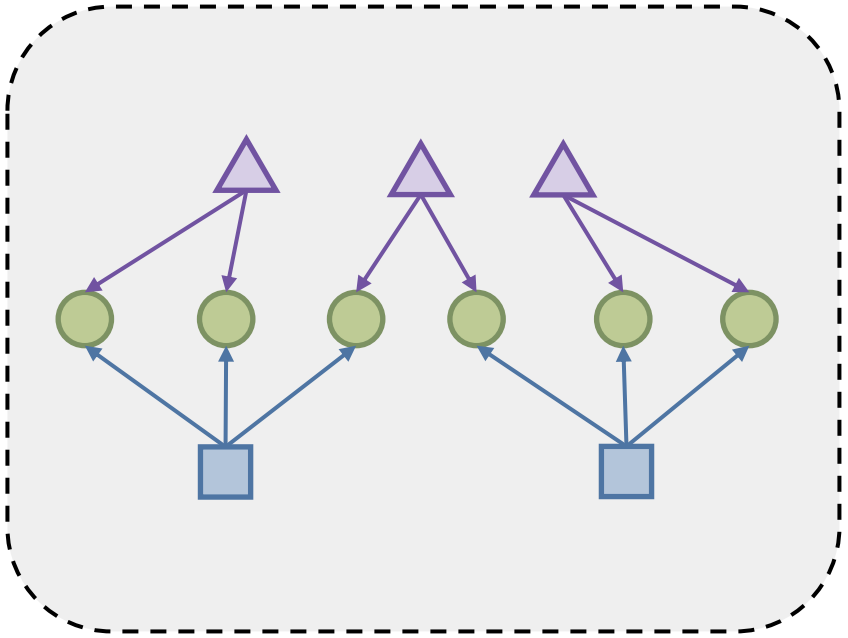
Multimodal Sequence Challenges

- Irregular lengths
- Misalignment
- Fusion of more than 2 modalities
- **Long-term** temporal dependency

Why Use Graph for Multimodal Sequence?



Why Use Graph for Multimodal Sequence?



Graph:

- Add nodes freely
- Build edges freely

Why Use Graph for Multimodal Sequence?

Challenges:

- Irregular lengths ✅
- Misalignment
- Fusion of more than 2 modalities
- Long-term temporal dependency

Graph:

- Add nodes freely
- Build edges freely

Why Use Graph for Multimodal Sequence?

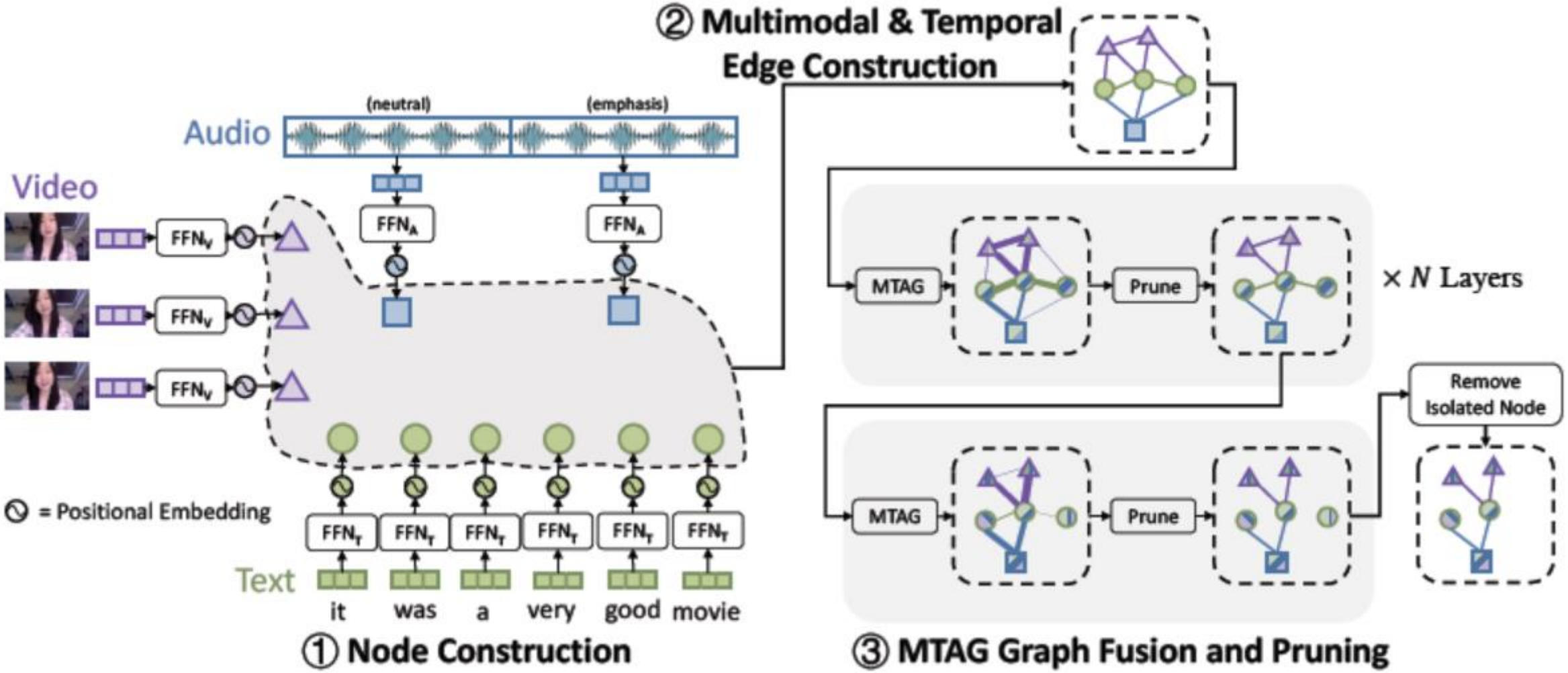
Challenges:

- Irregular lengths
- Misalignment ✅
- Fusion of more than 2 modalities ✅
- Long-term temporal dependency ✅

Graph:

- Add nodes freely
- Build edges freely

Method


② Multimodal & Temporal Edge Construction
(neutral)
(emphasis)
Audio
Video
FFN_A
FFN_V
FFN_T
= Positional Embedding
Text
it
was
a
very
good
movie
① Node Construction
MTAG
Prune
× N Layers
Remove Isolated Node
③ MTAG Graph Fusion and Pruning

Node construction

Step1. FFN -> same dimension

Step2. Position embedding -> encode temporal information

=>node v_i & identifier $\pi_i \in \{\text{Audio, Video, Text}\}$

Edge construction

1. For a given node of a particular modality, its interactions with nodes from **different modalities** should be considered differently
2. The **temporal order** of the nodes also plays a key role
   - Positive: (☹ → 😐 →☺)
   - Negative: (☺ → 😐 →☹)

=> By **indexing** edges with edge **types**, different modal and temporal interactions between nodes can be addressed separately.

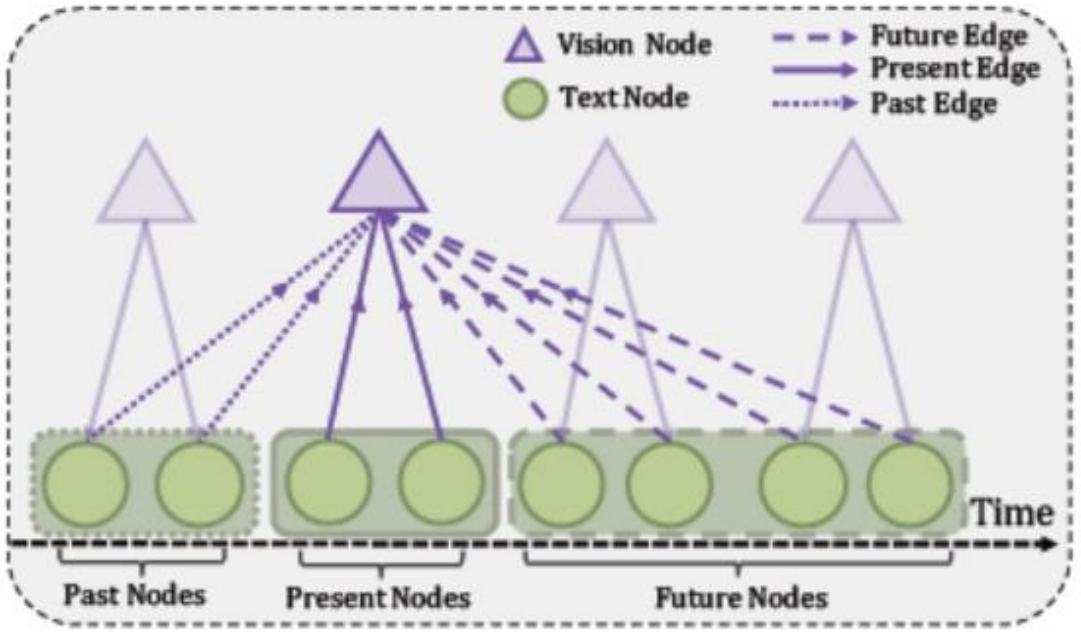
Multimodal Edges & Temporal Edges

- Multimodal Edges: assign e_{ij} with a modality type identifier $\phi_{ij} = (\pi_i \rightarrow \pi_j)$
- Temporal Edges: assign a temporal label τ_{ij} , {past, present, future}.
  - pseudo-alignment -> determine the temporal orders for nodes across different modalities

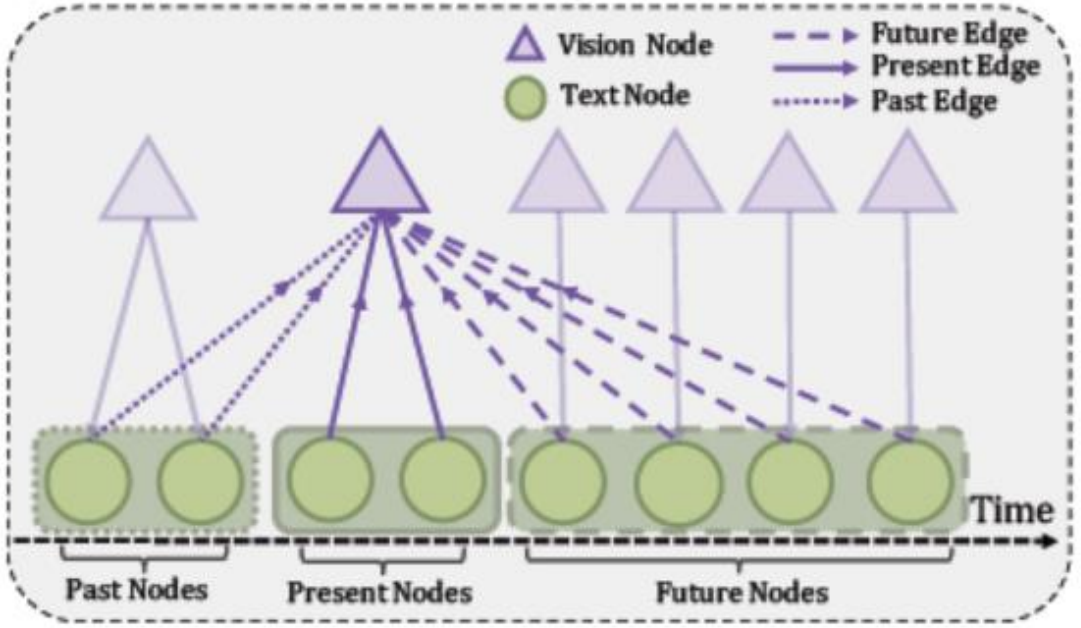
Pseudo-Alignment

- Input: longer sequence
- Output: shorter sequence
- Goal: find a feasible stride and kernel size that aligns the input and output

$$\begin{cases} S = \lceil \mathbf{avg}(1, \lfloor \frac{M}{N-1} \rfloor) \rceil, \\ W = M - (N-1) * S \\ S = 2, W = 2 \end{cases} \quad \begin{array}{l} \textbf{if } N \leq \frac{M}{2} \\ \\ \textbf{otherwise} \end{array}$$



(a) Pseudo-Alignment example with less vision nodes



(b) Pseudo-Alignment example with more vision nodes

Fusion & Pruning

- Raw attention score

$$\beta_{[h],i,j} = \text{LeakyRelu}(\mathbf{a}_{[h]}^{\phi_{ji},\tau_{ji}} \cdot [\mathbf{x}'_i \| \mathbf{x}'_j])$$

- Attention weight (Normallize & softmax)

$$\alpha_{[h],i,j} = \frac{\exp(\beta_{[h],i,j})}{\sum_{k \in \mathcal{N}_i} \exp(\beta_{[h],i,k})}$$

- Node feature aggregation

$$\mathbf{z}_i = \underset{h=1}{\overset{H}{\text{concat}}}(\sum_{j \in \mathcal{N}_i} \alpha_{[h],i,j} \mathbf{x}'_j)$$

Algorithm 1: MTAG with edge pruning

Feature transformation $\mathbf{x}'_i \leftarrow \mathbf{M}_{\pi_i} \mathbf{x}_i, \forall i$

for $h = 1 ... H$ **do**

  for $j \in \mathcal{N}_i \cup i$ **do**

    calculate raw attention score using modality- and temporal-edge-type specific parameters: $\beta_{[h],i,j} = \text{LeakyRelu}(\mathbf{a}_{[h]}^{\phi_{ji},\tau_{ji}} \cdot [\mathbf{x}'_i \| \mathbf{x}'_j])$

  normalize raw attention scores over $\mathcal{N}_i \cup i$ to get attention weight $\alpha_{[h],i,j}$

calculate node output feature $\mathbf{z}_i = \underset{h=1}{\overset{H}{\text{concat}}}(\sum_{j \in \mathcal{N}_i \cup i} \alpha_{[h],i,j} \mathbf{x}'_j)$

calculate average attention weight across all heads $\overline{\alpha}_{i,j} = \frac{1}{H} \sum_{h=1}^{H} (\alpha_{[h],i,j})$

sort $\overline{\alpha}_{i,j}$ and delete the edges with the smallest $k\%$ average attention weight from $\mathcal{N}_i \cup i$, obtaining $\mathcal{N}'_i$

return $\mathbf{z}_i, \mathcal{N}'_i \ \forall i$

Graph Readout

Step1. Averaging all the surviving nodes' output features into one vector

Step2. 3-layer MLP

=> final prediction

Experiments

Model \ Emotion	Happy	Sad	Angry	Neutral
(Unaligned) IEMOCAP Emotions.				
CTC + EF-LSTM	75.7	70.5	67.1	57.4
LF-LSTM	71.8	70.4	67.9	56.2
CTC + RAVEN	76.8	65.6	64.1	59.5
CTC + MCTN	77.5	71.7	65.6	49.3
MulT	81.9	74.1	70.2	59.7
MTAG (ours)	**86.0**	**79.9**	**76.7**	**64.1**

Table 2: F1 Scores on unaligned IEMOCAP. Higher is better.

Model \ Metirc	$Acc_7 \uparrow$	$Acc_2 \uparrow$	F1 ↑	MAE ↓	Corr ↑
(Unaligned) CMU-MOSI Sentiment					
CTC+EF-LSTM	31.0	73.6	74.5	1.078	0.542
LF-LSTM	33.7	77.6	77.8	0.988	0.624
CTC+MCTN	32.7	75.9	76.4	0.991	0.613
CTC+RAVEN	31.7	72.7	73.1	1.076	0.544
MulT	**39.1**	81.1	81.0	0.889	0.686
MTAG (ours)	38.9	**82.3**	**82.1**	**0.866**	**0.722**

Table 3: Results on unaligned CMU-MOSI. ↑ means higher is better and ↓ means lower is better.

Datasets: IEMOCAP , CMU-MOSI

Experiments

Model	# Parameters
MulT (previous SOTA)	2.24 M
MTAG (ours)	**0.14 M**

Table 4: Number of model parameters (M = Million).

Ablation Study

Ablation	$Acc_2 \uparrow$	F1 ↑	MAE ↓
Edge Types			
No Edge Types	82.4	82.5	0.937
Multimodal Edges Only	85.6	85.7	**0.859**
Temporal Edges Only	85.2	85.2	0.887
Pruning			
Random Pruning Keep 80%	75.5	74.5	1.080
No Pruning	84.7	84.7	0.908
Modalities			
Language Only	81.5	81.4	0.911
Vision Only	57.0	57.1	1.41
Audio Only	58.1	58.1	1.37
Vision, Audio	62.0	59.2	1.360
Language, Audio	85.9	85.7	0.915
Language, Vision	86.6	86.6	0.896
Full Model, All Modalities	**87.0**	**87.0**	**0.859**

Finding 1:

Adding modality and temporal specific edges improves performance

Ablation Study

Ablation	$Acc_2 \uparrow$	F1 ↑	MAE ↓
Edge Types			
No Edge Types	82.4	82.5	0.937
Multimodal Edges Only	85.6	85.7	**0.859**
Temporal Edges Only	85.2	85.2	0.887
Pruning			
Random Pruning Keep 80%	75.5	74.5	1.080
No Pruning	84.7	84.7	0.908
Modalities			
Language Only	81.5	81.4	0.911
Vision Only	57.0	57.1	1.41
Audio Only	58.1	58.1	1.37
Vision, Audio	62.0	59.2	1.360
Language, Audio	85.9	85.7	0.915
Language, Vision	86.6	86.6	0.896
Full Model, All Modalities	**87.0**	**87.0**	**0.859**

Finding 2:

Top-K% pruning improves performance;
Random pruning decreases performance

Ablation Study

Ablation	$Acc_2 \uparrow$	F1 ↑	MAE ↓
Edge Types			
No Edge Types	82.4	82.5	0.937
Multimodal Edges Only	85.6	85.7	**0.859**
Temporal Edges Only	85.2	85.2	0.887
Pruning			
Random Pruning Keep 80%	75.5	74.5	1.080
No Pruning	84.7	84.7	0.908
Modalities			
Language Only	81.5	81.4	0.911
Vision Only	57.0	57.1	1.41
Audio Only	58.1	58.1	1.37
Vision, Audio	62.0	59.2	1.360
Language, Audio	85.9	85.7	0.915
Language, Vision	86.6	86.6	0.896
Full Model, All Modalities	**87.0**	**87.0**	**0.859**

Finding 3:
Language is the most helpful modality

MTAG increases its performance as more modalities are provided

Summary of Contributions

- A new pipeline to model unaligned multimodal sequence data
- A new graph convolution operation called MTAG fusion
- State-of-the-art results on two datasets
- Much fewer model parameters than previous SOTA