Autoencoded Quantification Of Unsupervised Representative Shapelets

CHEN Ziyuan & CHEN Zhirong

15 July 2022

Contents

Symbol Table

- s – time series
- t – subsequences
- v – shapelets
- *(We're following the convention of Time2Graph. Sorry for the counter-intuitiveness.)*

- The phrase "t-v distance" will frequently appear

Problem Analysis

- *Time2Graph* as a **Finite State Machine**
- **State**: Shape of a subsequence
- **Finite**: Number of shapes should be finite
- **Machine**: Transfer a series into a *state transition diagram*
  - Nodes: States / Shapes / Shapelets
  - Edge weights: Probability of interstate transition

Traditional Ways to Extract Shapelets

1. Choose shapelets from all possible subsequences

   Those which performs well on classification tasks are regarded as potential shapelets

2. Learn shapelets

   Generate shapelets which can perform well on classification tasks

Core Idea: **Good shapelets are good classifiers.**

Running classification is time-consuming. **Let's avoid this!**

Our Approach v1

- Sampling & Unsupervised labelling
  - Map shapelets to subsequences, calculate t-v distances as labels
- Unsupervised training
  - **Contrastively** train an embedding network U to extract features
- Supervised labelling
  - Feed t and v into U, use the E.D. of **output vectors** as "distance" labels
- Supervised training
  - Concatenate a network (MLP.....) S after U, use the labels ^ for fine-tuning
- The model is then ready!
  - For each s, transfer its t into possibility vectors and construct the FSM Diagram

Our Approach v1: Problem

These labels for fine-tuning are unreliable!!!

- Supervised labelling
  - Feed t and v into U, use the E.D. of **output vectors** as "distance" labels
- Supervised training
  - Concatenate a network (MLP……) S after U, use the labels ^ for fine-tuning
- Turn to the classification task?
  - **Then we're retreating to the old inefficient path!**

Shapelet Based Model vs. Time2Graph

- **Shapelet based models** find “signature” subsequences
  - Shapelets should represent the time series classes
- Time2Graph uses shapelet transfer embedding
  - Shapelets should represent the subsequences accurately.
- It sounds subtle, but the intuition is……
  - Shapelets are used to **express the subsequences**
  - INSTEAD OF **classify the time series** directly.
  - So maybe “selection by classification” is not the optimal option!

Our Approach v2

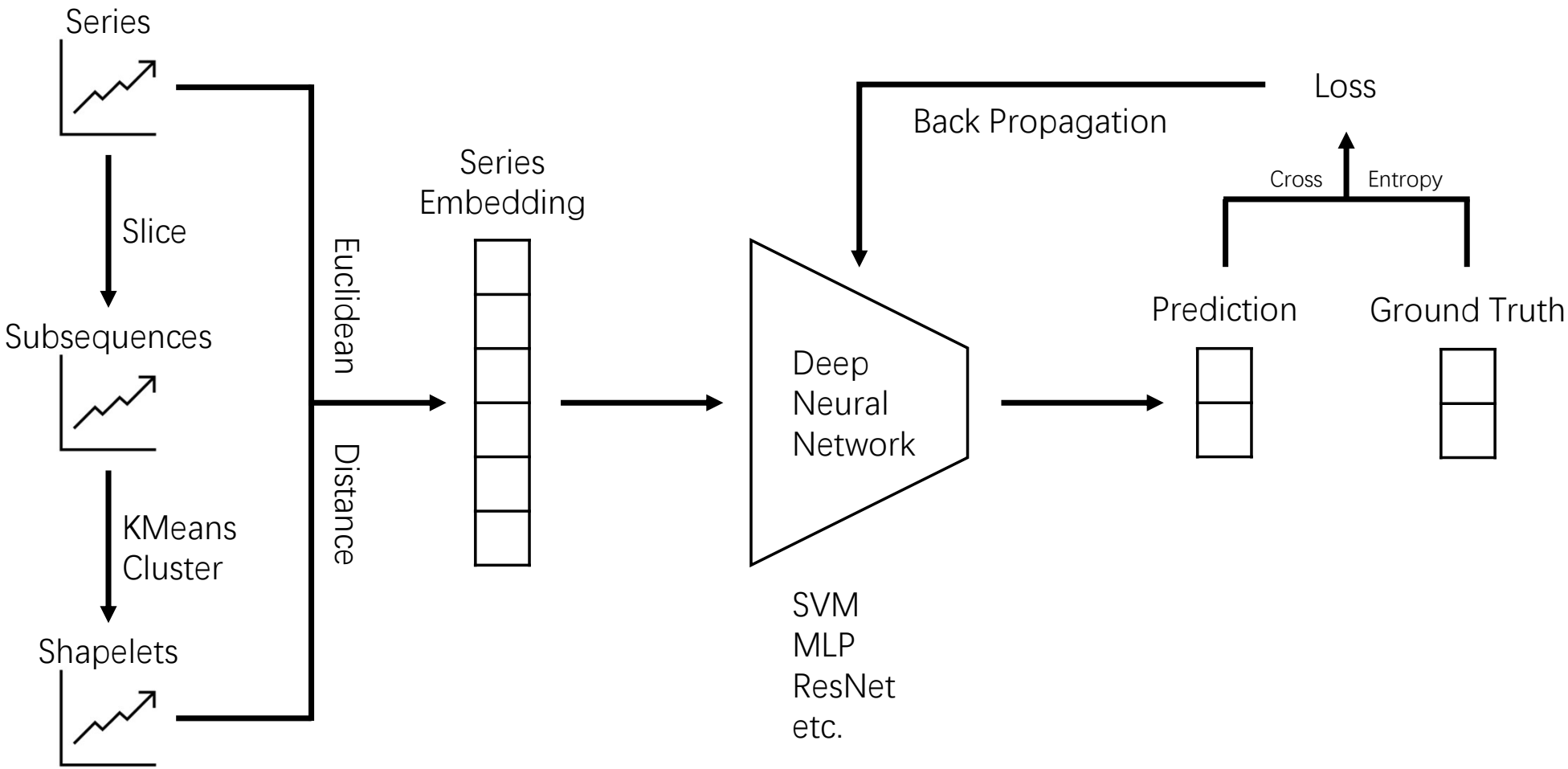
AIM FOR EXPRESSIVENESS
INSTEAD OF DISTINCTIVENESS

- The more **diverse**, the more expressive!
- Methodology: clustering
  - Use **KMeans** to cluster all subsequences
  - For each cluster, use the subseq. closest to the center as our “anchor”

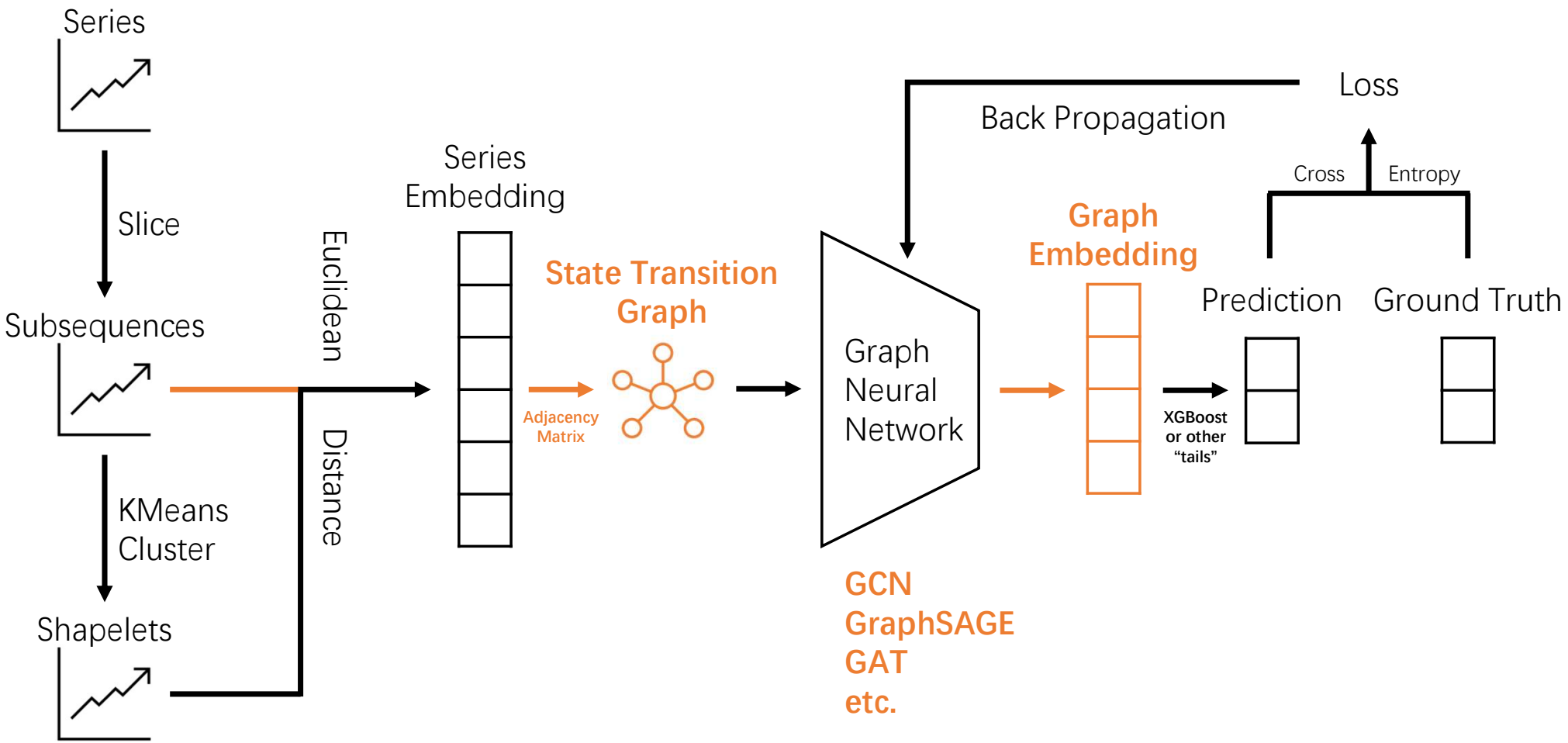
More on Picking the “Anchors”

- K-Means outperforms random picking
  - For each cluster, choose the subsequence closest to the center
  - Encourages representative and diverse shapelets
  - A mature Algorithm for speeding up – runs in O(n) time
  - ShapeNet shows the plausibility of selection via clustering
    - ^ Modified: choose the centroids in the nn-learned embedding space?

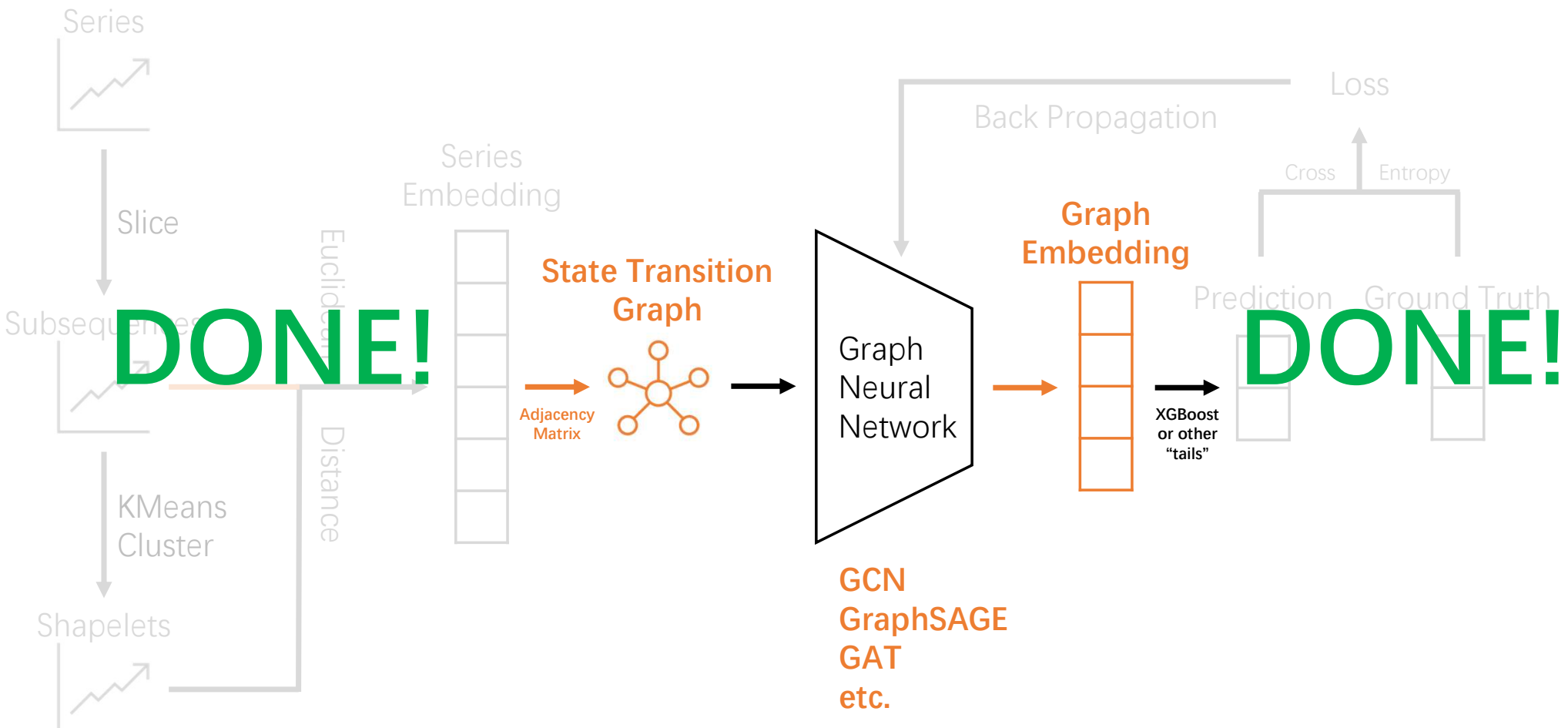
Model Diagram: SimpleKMeans



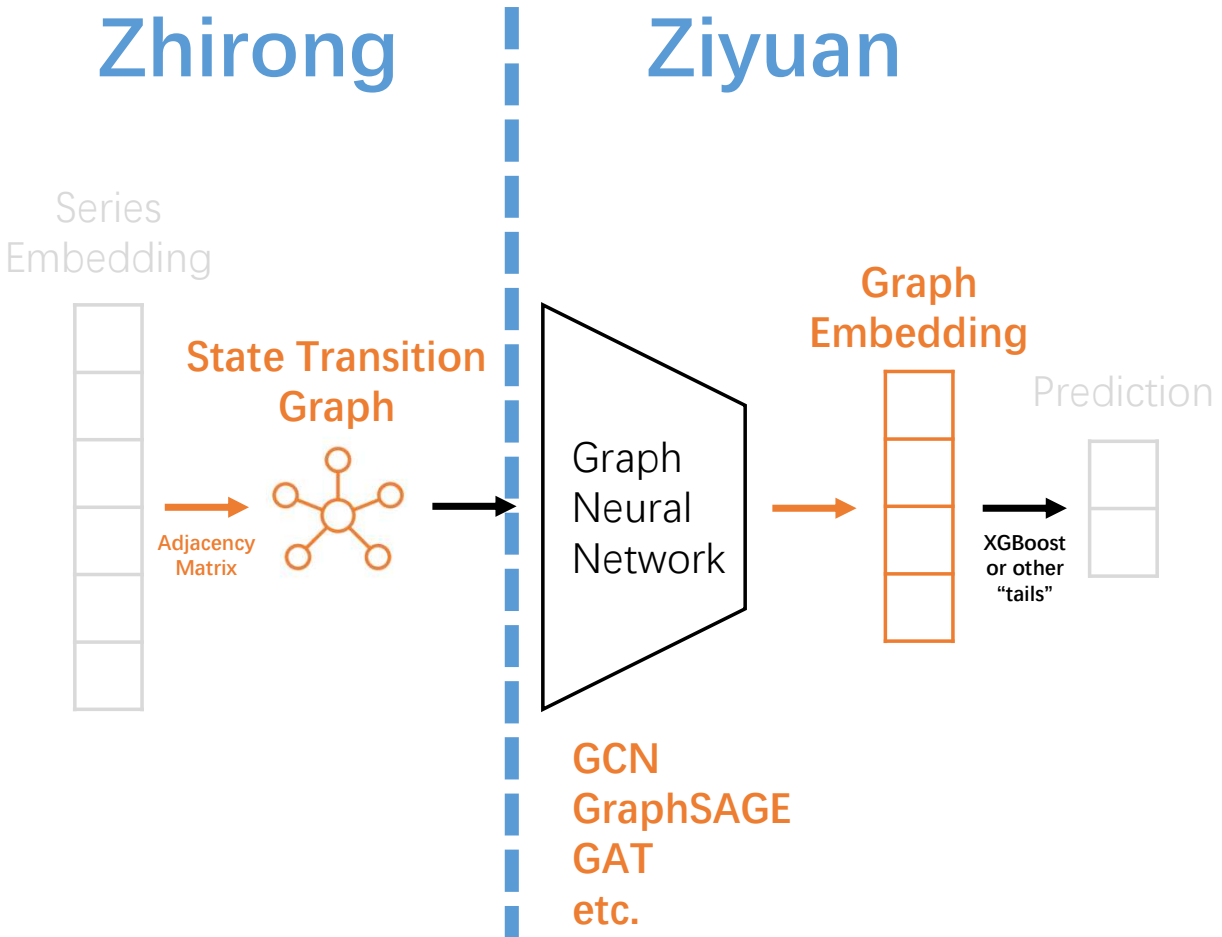
Model Diagram: Time2Graph



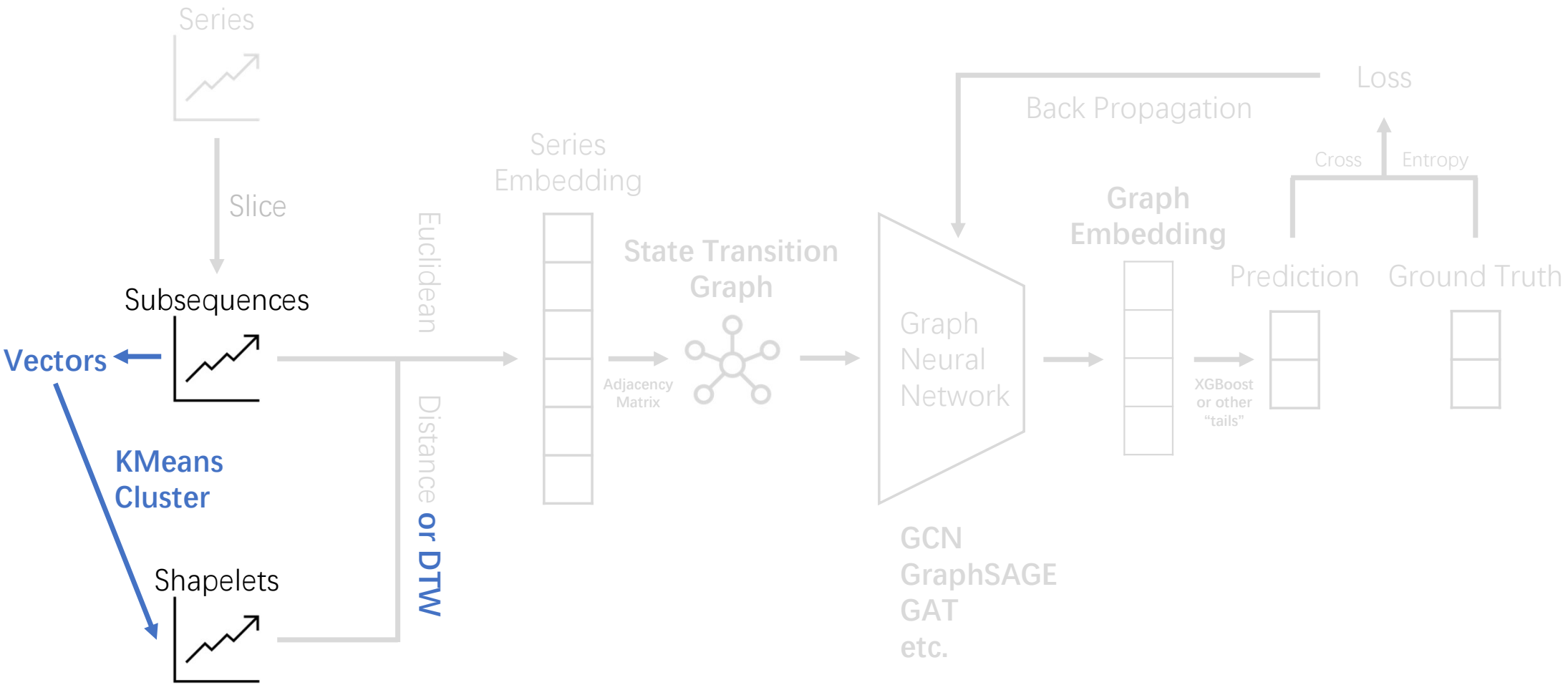
Last Week


Series
Slice
Subsequences
KMeans
Cluster
Shapelets
Euclidean
Distance
DONE!
Series
Embedding
State Transition
Graph
Adjacency
Matrix
Graph
Neural
Network
GCN
GraphSAGE
GAT
etc.
Graph
Embedding
XGBoost
or other
"tails"
Back Propagation
Loss
Cross
Entropy
Prediction
Ground Truth
DONE!

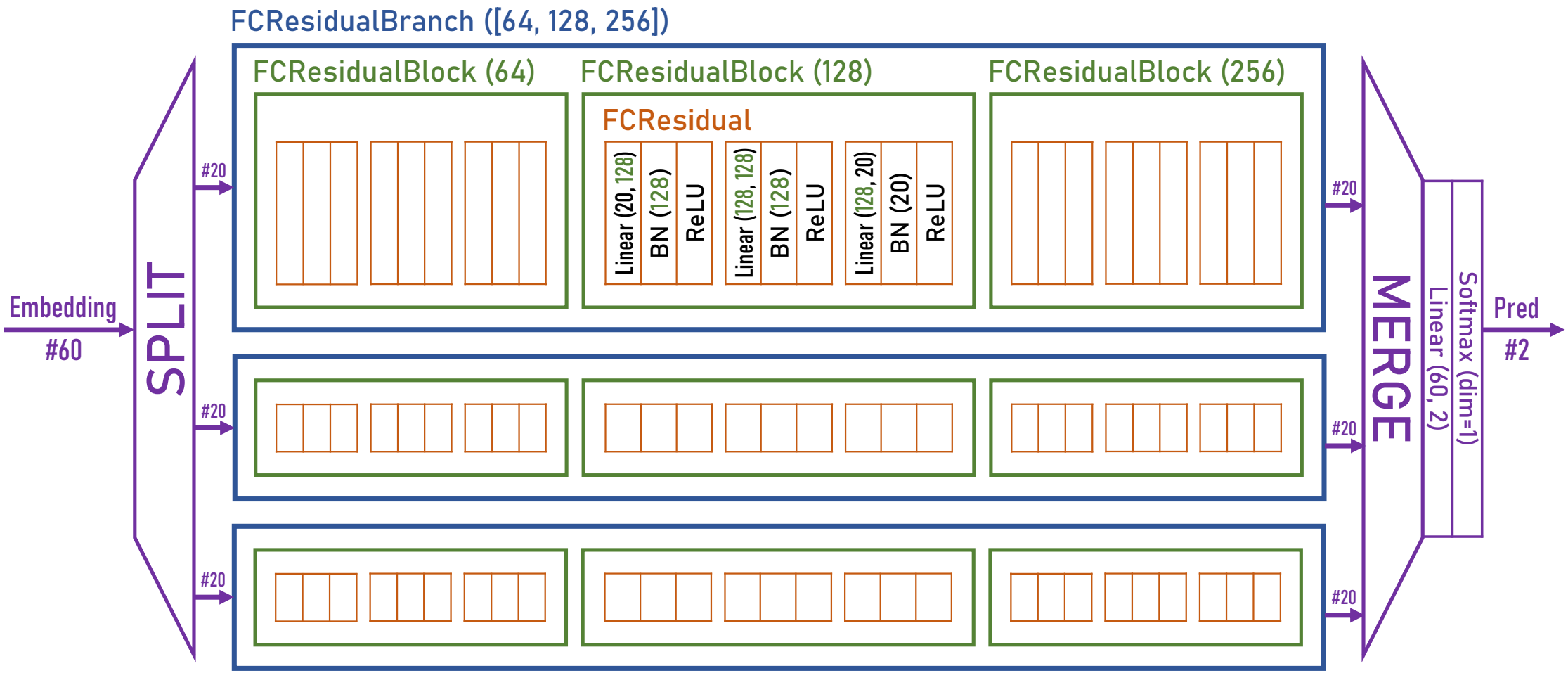
This Week


Zhirong
Ziyuan
Series Embedding
State Transition Graph
Adjacency Matrix
Graph Neural Network
Graph Embedding
Prediction
XGBoost or other "tails"
GCN
GraphSAGE
GAT
etc.

Model Diagram: AQOURSNet



Model Diagram: Hierarchical Networks



Experiments (Top 3 / Bottom 7 among 16 models)

Best Performance

Why do they fail???

Dataset	RotF	DTW_Rn	ST	LS	FS	SD	COTE	ELIS	ResNet	Random	BSPCOVER	KMeans_SVM	KMeans_MLP	KMeans_Res	KMeans_MLPHier	KMeans_ResHier
BeetleFly	90.00	65.00	90.00	80.00	70.00	75.00	80.00	85.00	85.00	80.00	90.00	95.00	*100.00*	95.00	50.00	*100.00*
Coffee	*100.00*	*100.00*	96.43	*100.00*	92.86	96.10	*100.00*	96.43	*100.00*	93.47	*100.00*	*100.00*	*100.00*	*100.00*	53.57	96.43
DistalPhalanxOutlineCorrect	75.72	72.46	77.54	77.90	75.00	71.70	76.09	57.83	77.10	77.52	*83.17*	77.90	80.07	78.62	58.33	73.19
Earthquakes	74.82	72.66	74.10	74.10	70.50	63.60	74.82	77.64	71.20	75.49	*81.68*	74.82	74.82	78.42	74.82	74.82
ECG200	85.00	88.00	83.00	88.00	81.00	81.80	88.00	80.00	87.40	85.00	*92.00*	83.00	64.00	91.00	64.00	64.00
ECGFiveDays	90.82	79.67	98.37	*100.00*	99.77	95.30	99.88	95.45	97.50	89.95	*100.00*	96.28	93.96	97.68	----	85.02
FordA	84.47	66.52	*97.12*	95.68	78.71	77.60	95.68	67.60	92.00	90.12	96.31	75.53	52.88	65.45	51.59	65.61
Ham	71.43	60.00	68.57	66.67	64.76	61.90	64.76	63.81	75.70	71.87	*76.19*	71.43	72.38	73.33	51.43	66.67
ShapeletSim	41.11	69.44	95.56	95.00	*100.00*	67.20	96.11	*100.00*	77.90	79.25	84.44	*100.00*	82.22	96.11	53.33	87.78
SonyAIBORobotSurface1	80.87	69.55	84.36	81.03	68.55	85.00	84.53	87.85	*95.80*	80.06	88.35	92.18	92.35	78.87	57.07	68.22
SonyAIBORobotSurface2	80.80	85.94	93.39	87.51	79.01	78.00	95.17	93.17	*97.80*	79.93	93.49	91.71	84.78	92.97	61.70	75.55
Strawberry	97.30	94.59	96.22	91.08	90.27	88.40	95.14	83.85	*98.10*	89.57	94.29	93.24	86.49	96.76	64.32	83.51
ToeSegmentation1	53.07	75.00	96.49	93.42	95.61	88.20	97.37	*98.24*	96.30	81.71	96.49	92.11	90.79	94.74	55.26	80.70
TwoLeadECG	97.01	86.83	99.74	99.65	92.45	86.70	99.30	99.82	*100.00*	92.19	99.65	96.14	92.80	92.01	50.04	78.58
Wafer	99.45	99.59	*100.00*	99.61	99.68	99.30	99.98	99.43	99.90	96.49	99.81	98.91	89.21	99.43	89.21	89.21
WormsTwoClass	68.83	58.44	*83.12*	72.73	72.73	64.10	80.52	71.82	74.70	71.24	74.59	81.82	81.82	*83.12*	57.14	76.62
Yoga	82.43	84.30	81.77	83.43	69.50	62.50	87.67	83.90	87.00	81.27	*88.20*	75.23	73.30	85.07	53.57	65.33

Experiments (Top / Middle / Bottom)

Model	KMeans_SVM		KMeans_MLP		KMeans_ResNet	
Best Parameters	Number	Length	Number	Length	Number	Length
BeetleFly	30	30%	30	15%	20	20%
Coffee	20	10%	25	20%	20	25%
DistalPhalanxOutlineCorrect	30	20%	30	30%	30	15%
Earthquakes	10	20%	25	20%	25	15%
ECG200	30	30%	20	25%	30	30%
ECGFiveDays	30	30%	25	30%	25	30%
FordA	30	20%	20	25%	20	15%
Ham	30	20%	25	20%	25	30%
ShapeletSim	10	5%	30	15%	20	15%
SonyAIBORobotSurface1	20	20%	30	25%	25	25%
SonyAIBORobotSurface2	30	10%	25	20%	30	15%
Strawberry	20	5%	20	25%	30	25%
ToeSegmentation1	30	20%	30	15%	30	30%
TwoLeadECG	20	30%	25	30%	20	30%
Wafer	30	30%	25	25%	30	30%
WormsTwoClass	30	10%	30	15%	30	20%
Yoga	30	10%	30	15%	30	20%

Method	EQS	WTC	STB
RotF	74.82	***97.30***	68.83
DTW_Rn	72.66	94.59	58.44
ST	74.10	96.22	83.12
SD	63.60	88.40	64.10
COTE	74.82	95.14	80.52
ELIS	77.64	83.85	71.82
ResNet	71.20	***98.10***	74.70
Random	75.49	89.57	71.24
BSPCOVER	***81.68***	94.29	74.59
NN-ED	68.22	62.41	95.60
NN-DTW	70.31	68.16	95.53
NN-WDTW	69.50	67.74	95.44
NN-CID	69.41	69.56	95.51
DDTW	70.79	70.92	95.60
XGBoost Origin	74.82	62.34	95.92
XGBoost Feature	75.54	64.94	***97.03***
BoP	74.80	74.42	96.45
TSF	74.67	68.51	96.27
EE	73.50	71.74	95.88
SAXVSM	73.76	72.10	***96.97***
LS	74.22	73.57	92.49
FS	74.66	70.58	91.66
LPS	66.78	74.26	96.35
MLP	70.29	59.86	96.58
LSTM	74.82	42.86	63.84
VAE	71.22	62.34	71.35
Shapelet-Seq	75.53	55.84	78.10
Time2Graph	***79.14***	72.73	***96.76***
Time2Graph+ Static	76.98	70.13	95.95
Time2Graph+	77.70	71.43	96.49
KMeans_SVM	74.82	93.24	81.82
KMeans_MLP	74.82	86.49	81.82
KMeans_ResNet	***78.42***	***96.76***	83.12

Next Steps

- XGBoost implementation
  - Joint learning with GAT (PyTorch)
- Parameter adjustments
  - Shapelet embedding: `num_shapelets`, `num_segments`, `pruning_percentile`
  - Graph embedding: `hidden_dim`, `embed_dim`, `num_layers`, `heads`, `neg_slope`, `dropout`
  - Training options: `epochs`, `lr`, `weight_decay`
- Time2Vec & DTW functionality verification
  - Current `ts2vec` is slow……