Learning to Embed Time Series Patches Independently

Seunghan Lee[1], *Tayeoung Park*[1,2] , *Kibok Lee*[1,2]

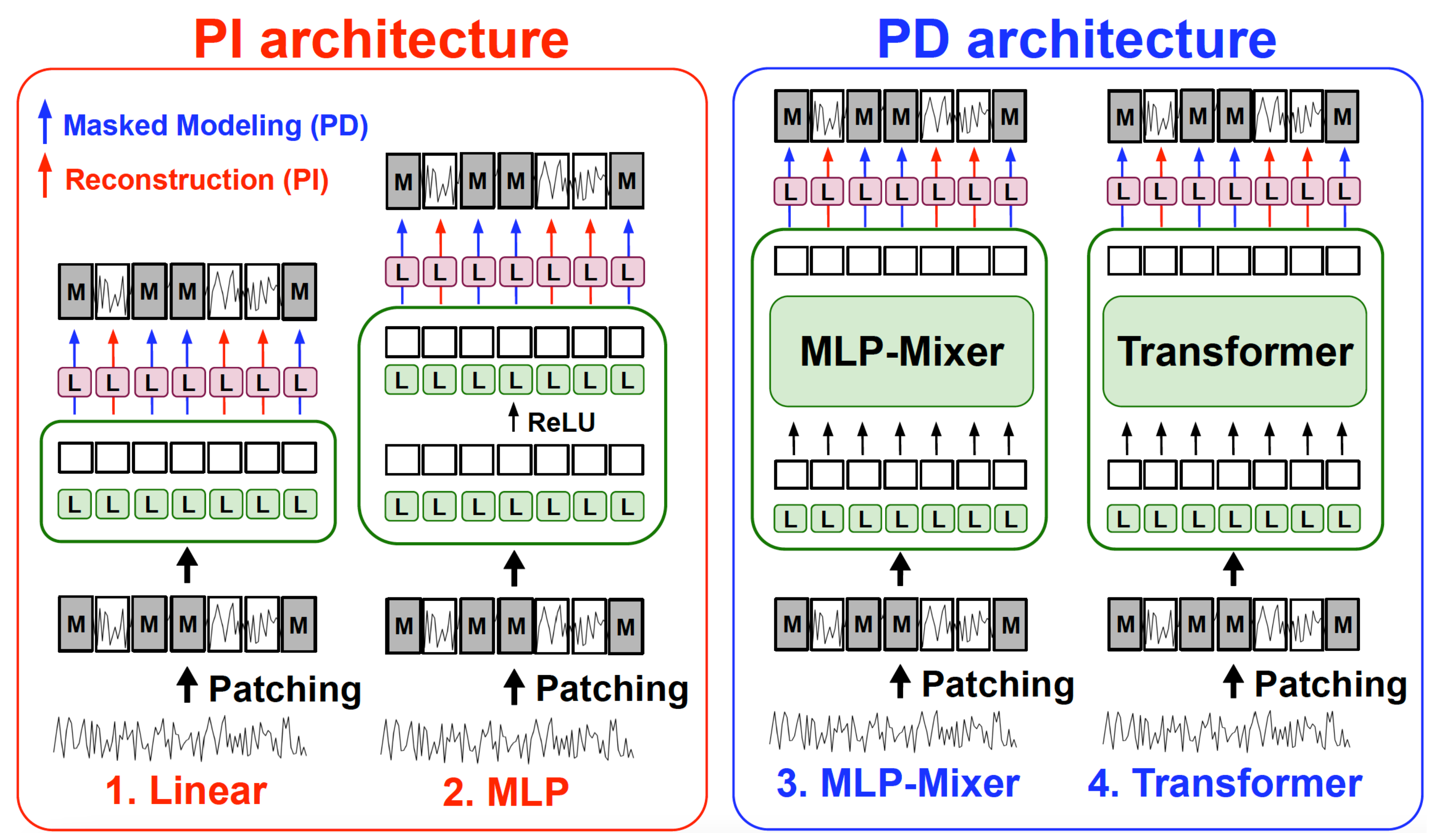
[1]*Department of Statistics and Data Science, Yonsei University,* [2]*Department of Applied Statistics, Yonsei University*

1. Self-Supervised Learning in Time Series (TS)

- **Masked Time-series Modeling**: masking patches within a TS and predicting the masked values
- **Contrastive Learning**: maximizing similarities between positive pairs while minimizing similarities between negative pairs

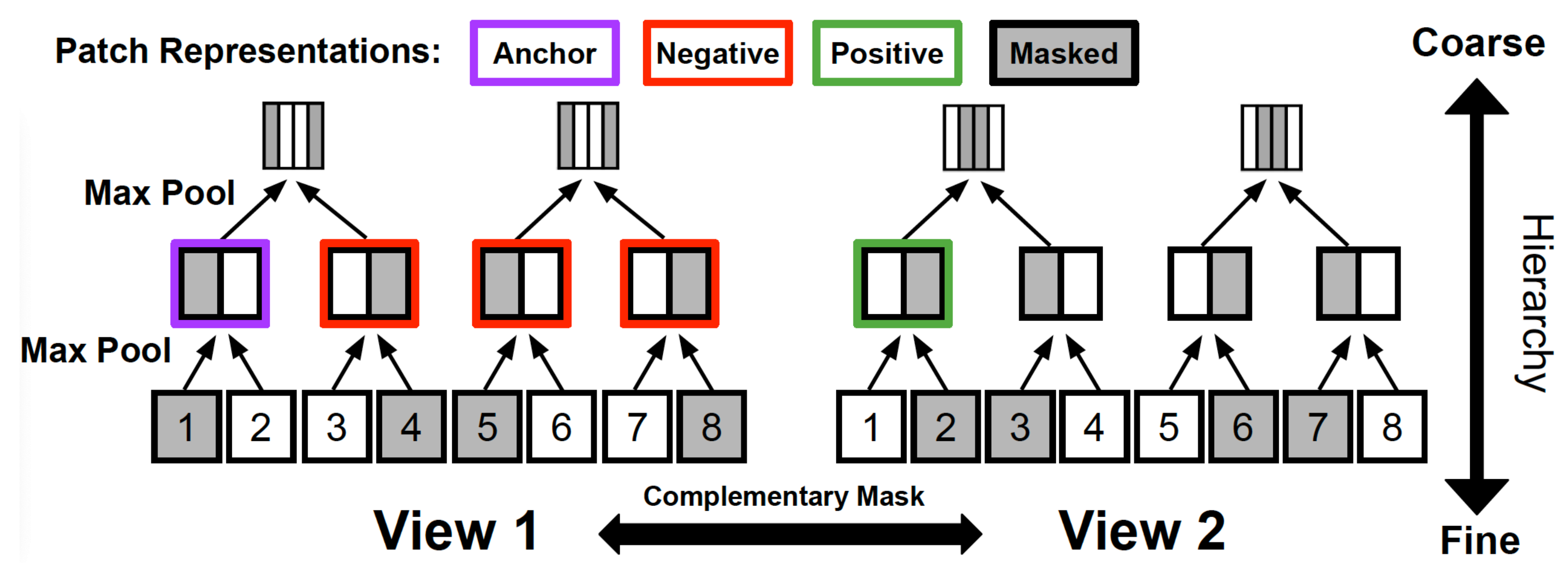
2. Patch Independent Task & Architecture

- **Patch Independence (PI)**: does not consider an interaction btw patches when embedding them
- **Patch Dependence (PD)**: consider an interaction btw patches when embedding them



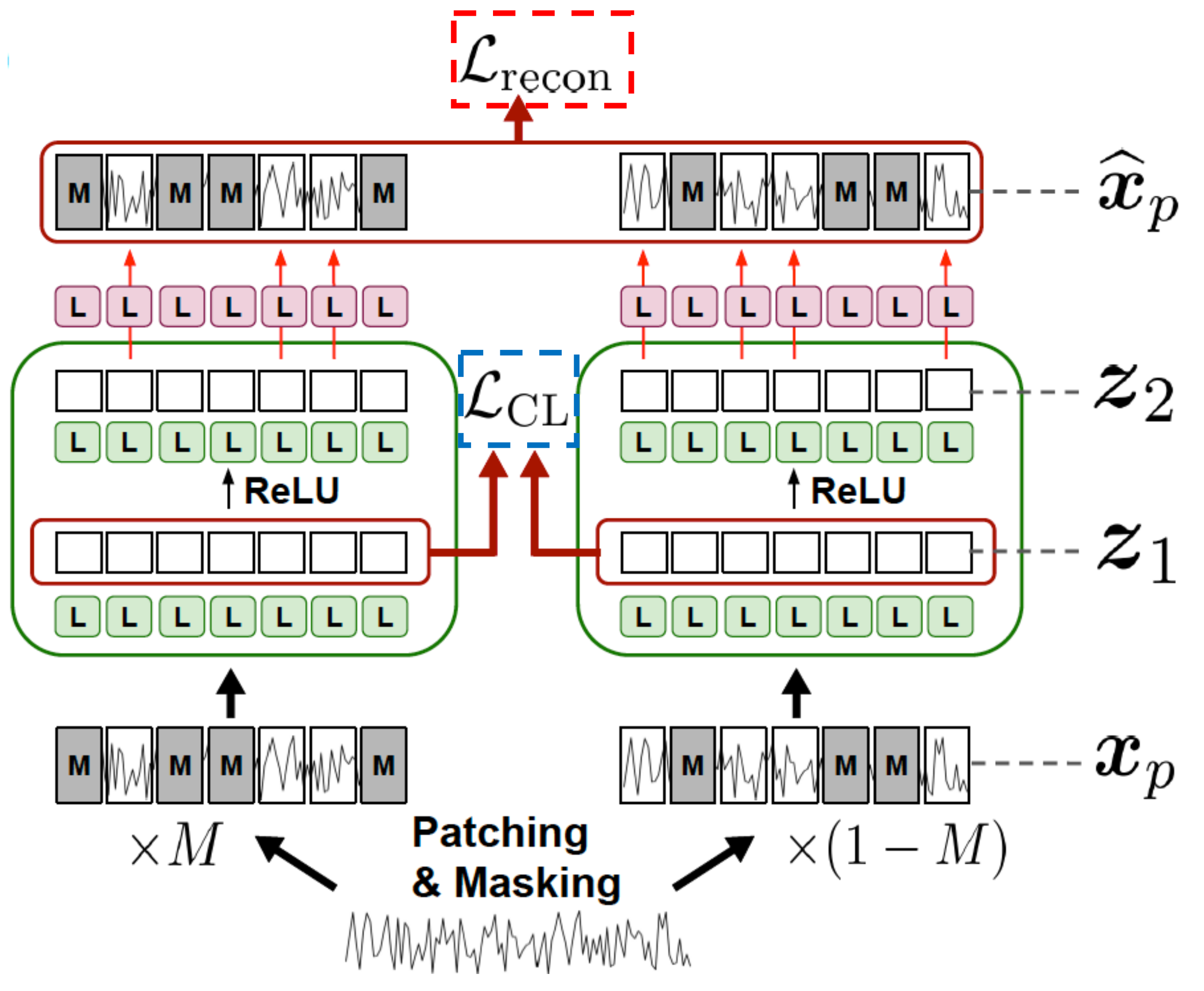
3. Complementary Contrastive Learning (CL)

- Propose complementary CL to hierarchically capture adjacent TS information efficiently
- Adopt the hierarchical contrastive loss proposed in TS2Vec (Yue et al., 2022)
- Losses are computed on intermediate representations after each max-pooling layer along the temporal axis and then aggregated



4. Patch Independence for TS (PITS)

- PI encoder: **2-layer MLP**
- PI pretrain task: **reconstruction task**



For conciseness, let the two views as $\boldsymbol{z}_1^{(i,c,n+N)}$ & $\boldsymbol{z}_1^{(i,c,n)} = \boldsymbol{z}_1^{(i,c,n+2N)}$
$\boldsymbol{x}_p = \left\{\boldsymbol{x}_p^{(i,c,n)}\right\}$, $\boldsymbol{z} = \left\{\boldsymbol{z}^{(i,c,n)}\right\}$, and $i = 1,\ldots,B$, $c = 1,\ldots,C$, $n = 1,\ldots,N$

(1) Reconstruction Loss

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{B}\sum_{c=1}^{C}\sum_{n=1}^{N}\left\|\boldsymbol{m}^{(i,c,n)} \odot \left(\boldsymbol{x}_p^{(i,c,n)} - \widehat{\boldsymbol{x}}_p^{(i,c,n)}\right)\right\|_2^2 + \left\|\left(1-\boldsymbol{m}^{(i,c,n)}\right) \odot \left(\boldsymbol{x}_p^{(i,c,n)} - \widehat{\boldsymbol{x}}_p^{(i,c,n)}\right)\right\|_2^2$$
$$= \sum_{i=1}^{B}\sum_{i=1}^{C}\sum_{n=1}^{N}\left\|\boldsymbol{x}_p^{(i,c,n)} - \widehat{\boldsymbol{x}}_p^{(i,c,n)}\right\|_2^2,$$

(2) Contrastive Loss

$$p(i,c,(n,n')) = \frac{\exp(\boldsymbol{z}_1^{(i,c,n)} \circ \boldsymbol{z}_1^{(i,c,n')})}{\sum_{s=1,s\neq n}^{2N}\exp(\boldsymbol{z}_1^{(i,c,n)} \circ \boldsymbol{z}_1^{(i,c,s)})}$$

$$\mathcal{L}_{\text{CL}} = \frac{1}{2BCN}\sum_{i=1}^{B}\sum_{i=1}^{C}\sum_{n=1}^{2N} -\log p(i,c,(n,n+N))$$

$$\boldsymbol{z}^{(i,c,n)} \leftarrow \text{MaxPool}([\boldsymbol{z}^{(i,c,2n-1)}, \boldsymbol{z}^{(i,c,2n)}]), \quad N \leftarrow \lfloor N/2 \rfloor$$

(3) Total Loss

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{CL}}$$

5. Effectiveness of PI strategies

(1) Pretraining with PI task consistently outperforms the PD task across all architectures

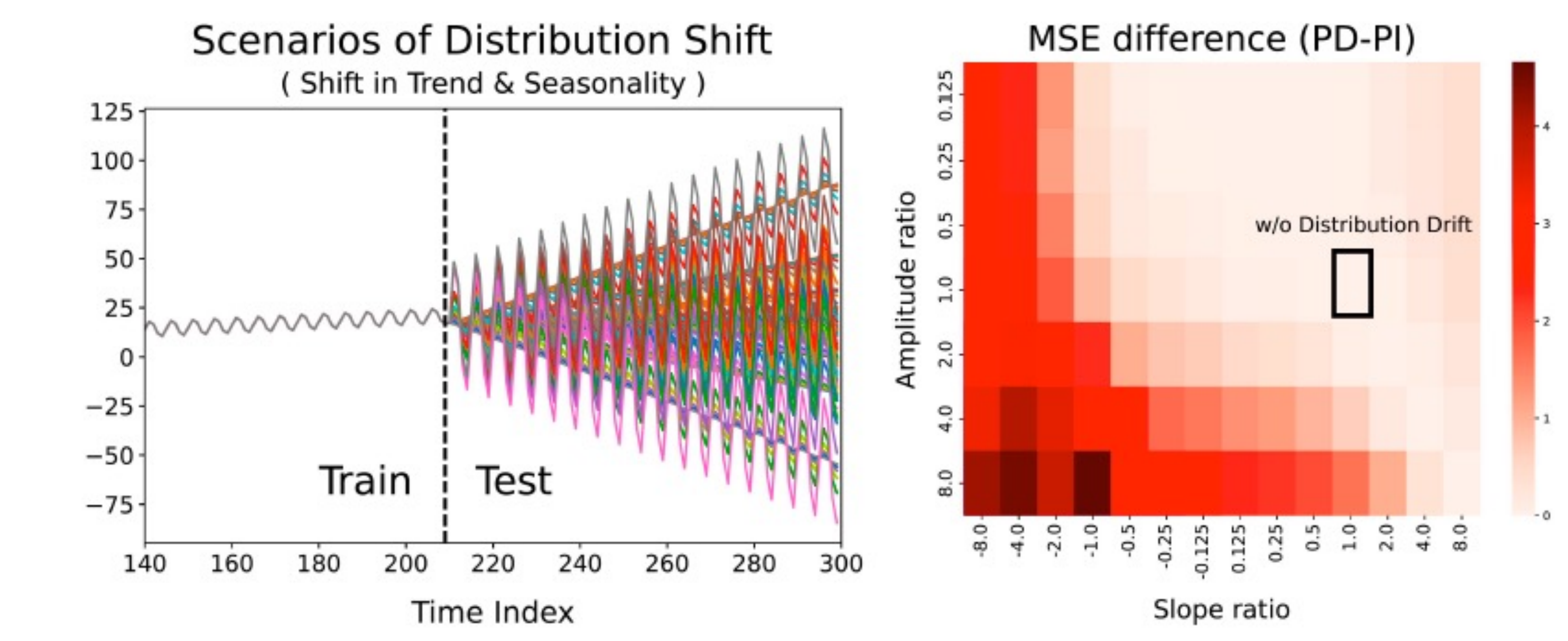
	PI acrhitecture						PD architecture					
	Linear			MLP			MLP-Mixer			Transformer		
Task	PD	PI	Gain(%)	PD	PI	Gain(%)	PD	PI	Gain(%)	PD	PI	Gain(%)
ETTh1	0.408	0.408	+0.0	0.418	0.407	**+2.6**	0.420	0.409	**+2.6**	0.425	0.415	**+2.4**
ETTh2	0.343	0.338	**+1.5**	0.361	0.334	**+7.5**	0.365	0.341	**+6.6**	0.353	0.342	**+3.1**
ETTm1	0.359	0.358	**+0.2**	0.356	0.355	**+0.3**	0.354	0.352	**+0.6**	0.350	0.350	+0.0
ETTm2	0.254	0.243	**+0.4**	0.258	0.253	**+1.9**	0.259	0.253	**+2.3**	0.274	0.256	**+6.6**
Average	0.342	0.340	**+0.3**	0.348	0.337	**+3.2**	0.350	0.339	**+3.1**	0.351	0.341	**+2.8**

(2) Comparison with PatchTST in terms of the # of params & training/inference time

- Dataset: ETTm2

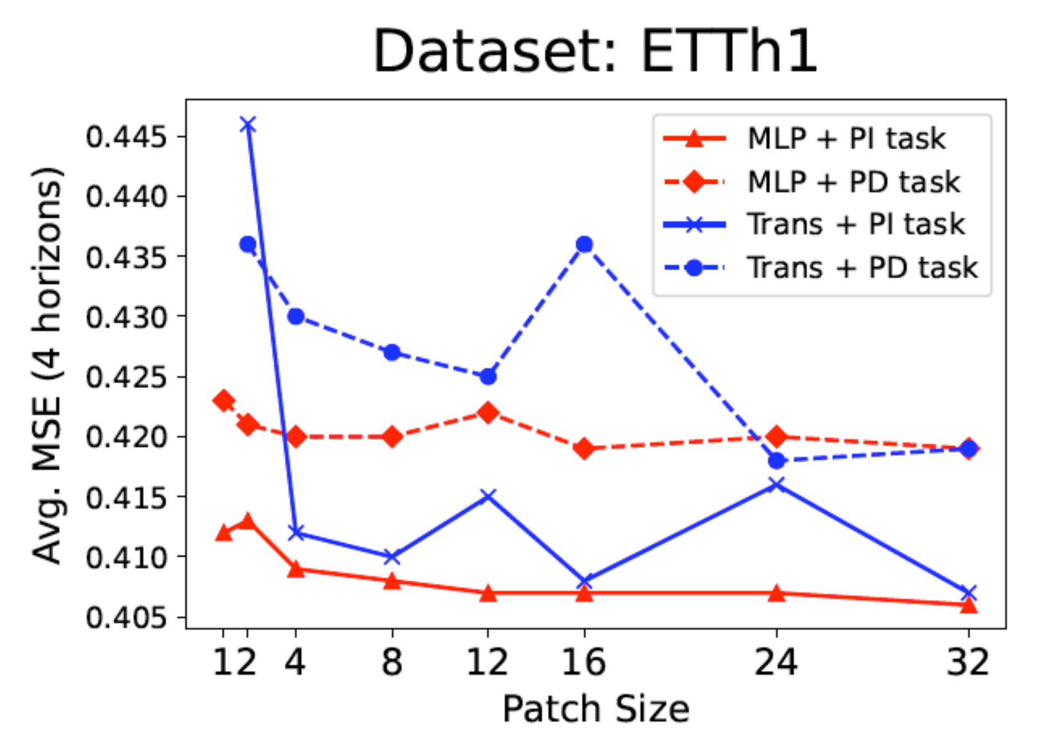
	Self-supervised settings			
	PatchTST	PITS		
		w/o CL	w/ CL	w/ hier. CL
Number of params	406,028	5,772		
Pretrain time (min)	77	15	17	25
Inference time (sec)	7.5	3.3		
Avg. MSE	0.274	0.253	0.252	0.244

(3) Robustness to distribution shift



- (Left) 98 toy TS exhibiting varying degrees of distribution shift by changing slope & amplitude
- (Right) MSE difference (= MSE of PD task – MSE of PI task) of 98 datasets

(4) Robustness to patch size



References

Yue, Zhihan, et al (2020). "Ts2vec: Towards universal representation of time series." In AAAI 2022

Nie, Yuqi, et al. (2023) "A time series is worth 64 words: Long-term forecasting with transformers." In ICLR 2023