BRL Seminar

(2024. 07. 10. Wed)

Foundation model for Time Series

통합과정 8학기 이승한

Contents

Gao, Shanghua, et al. "Units: Building a unified time series model." *arXiv preprint arXiv:2403.00131* (2024).

1. Preliminaries

1. Preliminaries

1-1. Foundation Model

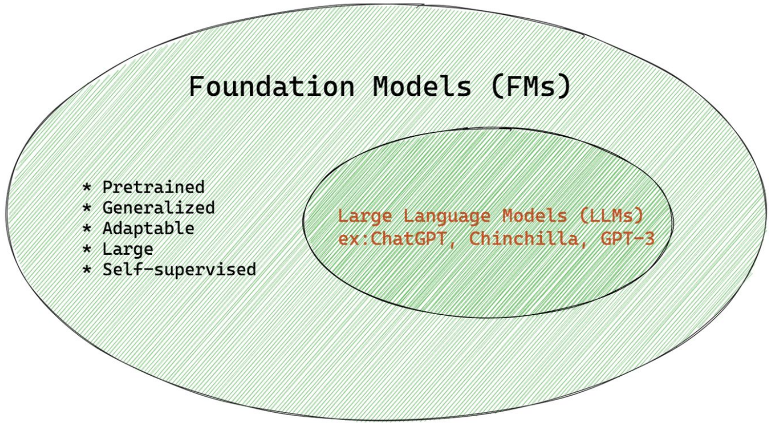
1-2. Characteristics of Time Series (TS)

1-3. TS Foundation Model

1. Preliminaries

1-1. Foundation Model (FM)

- **Large-scale**, **pre-trained** models
    - Serve as a base for various downstream tasks
- Pretrained on **vast amounts of data**
    - Often using self-supervised learning
    - Learn wide range of information & patterns
- **Strong generalization** ability
- Adaptable to **multiple tasks**
    - w/o task-specific training from scratch
    - via fine tuning / prompt tuning
- Examples: GPT-3, BERT, and DALL-E



FMs are models trained on broad data (using self-supervision at scale)
that can be adapted to a wide range of downstream tasks.
https://hai.stanford.edu/news/reflections-foundation-models

1. Preliminaries

1-2. Characteristics of Time Series (TS)

Heterogeneous in terms of…

- (1) **Explicit** characteristics
  - 1-a) Length (= L)
  - 1-b) Number of channels (= C)
- (2) **Implicit** characteristics
  - i.e. Nature of temporal dependencies, seasonality, trend …

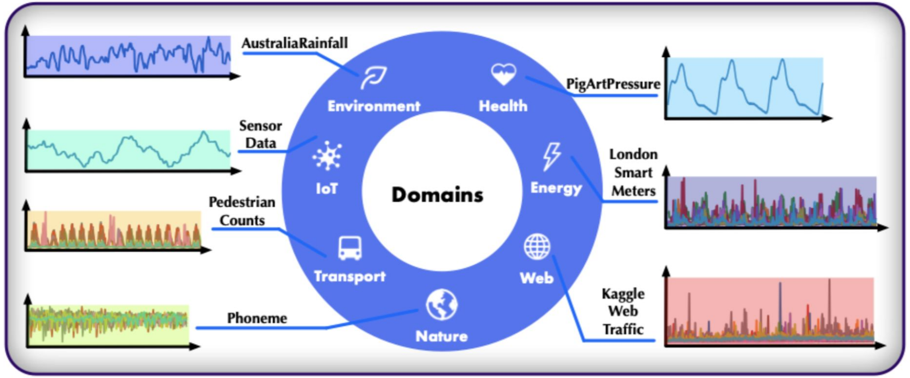
*Q) Can we make a **unified model** with such multiple **heterogeneous** TS datasets?*

1. Preliminaries

1-3. TS Foundation Model

Challenges of foundation model in TS

- # 1) **Heterogeneous TS**
  - Different C & L -> How to consider them in a unified architecture?
- # 2) **Lack of large-scale datasets**
  - Much smaller, compared to Vision & NLP



Diverse Shapes, Patterns and Domains

1. Preliminaries

1-3. TS Foundation Model

Categories of TS Foundation Models (2023~)

- (1) Based on **LLMs**
  - Due to challenge # 2)
  - Use pretrained model from NLP domain
- (2) Based on **TS** itself
  - Overcome challenge #2)
  - Construct large-scale TS dataset & pretrain with them

1. Preliminaries

1-3. TS Foundation Model

Categories of TS Foundation Models (2023~)

- (1) Based on **LLMs**
  - Due to challenge # 2)
  - Use pretrained model from NLP domain
- (2) Based on **TS** itself
  - Overcome challenge #2)
  - Construct large-scale TS dataset & pretrain with them

Recent trends

Timer	ICML 2024	Timer: Generative Pre-trained Transformers Are Large Time Series Models
TimeSiam	ICML 2024	TimeSiam: A Pre-Training Framework for Siamese Time-Series Modeling
Moirai	ICML 2024	Unified Training of Universal Time Series Forecasting Transformers
MOMENT	ICML 2024	MOMENT: A Family of Open Time-series Foundation Models
GTT	Arxiv 2024	Only the Curve Shape Matters: Training Foundation Models for Zero-Shot Multivariate Time Series Forecasting through Next Curve Shape Prediction
UniST	KDD 2024	UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction
TOTEM	Arxiv 2024	TOTEM: TOkenized Time Series EMbeddings for General Time Series Analysis
GPHT	Arxiv 2024	Generative Pretrained Hierarchical Transformer for Time Series Forecasting
UniTS	Arxiv 2024	UniTS: Building a Unified Time Series Model
SimTS	ICASSP 2024	Rethinking Contrastive Representation Learning for Time Series Forecasting
UniCL	Arxiv 2024	UniCL: A Universal Contrastive Learning Framework for Large Time Series Models
NuwaTS	Arxiv 2024	NuwaTS: a Foundation Model Mending Every Incomplete Time Series

2. UNITS: A Unified Multi-Task Time Series Model

(arxiv 2024)

2. UNITS

2-1. Abstract

Foundation models (FM): **“single”** pretrained model to **“multiple”** tasks

- via few shot prompting or fine-tuning

Lack of FM research in TS domain, due to …

- (1) **Inherent diverse & multi-domain TS**
- (2) **Diverging task specifications**
  - i.e. goal of TS forecasting != TS classification

=> Propose ***UNITS (Unified TS model)***

2. UNITS

2-1. Abstract

UniTS (Unified TS model)

- Supports **universal task specification**
  - (1) classification (CLS)
  - (2) forecasting (FCST)
  - (3) imputation (IMP)
  - (4) anomaly detection (AD)
- Novel **unified** network backbone
  - **Sequence & Variable attention**
    - Sequence = "**INTRA**-series" relation
    - Variable = "**INTER**-series" relation

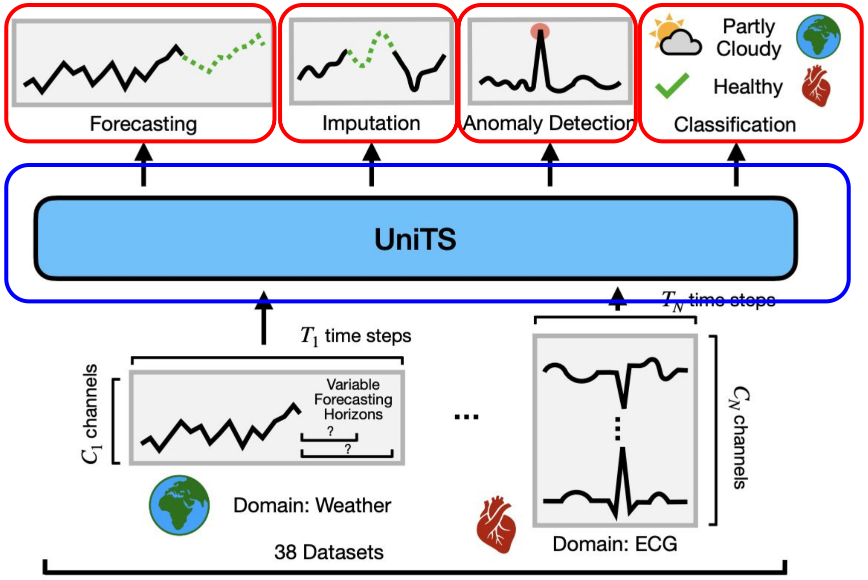


Figure 1: UNITS is a unified multi-task time series model that can process predictive and generative tasks across time series domains.

2. UNITS

2-1. Abstract

UniTS (Unified TS model)

Experiments

- a) Dataset) **38 multi-domain datasets**
- b) Performance) superior, compared to ..
  - baseline 1) task-specific models
  - baseline 2) LLM-based FM models
- c) Others
  - remarkable zero-shot, few-shot, prompt learning capabilities on new data & tasks

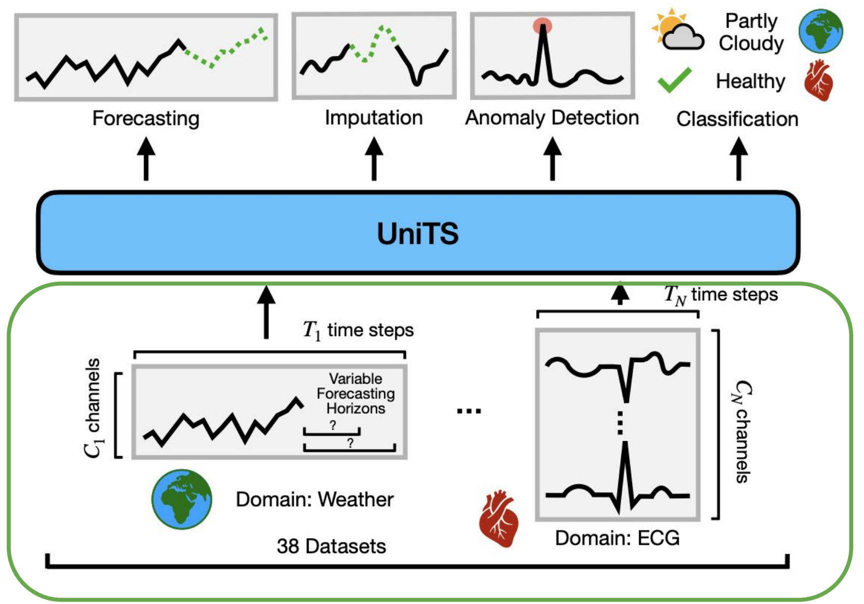


Figure 1: UNITS is a unified multi-task time series model that can process predictive and generative tasks across time series domains.

2. UNITS

2-2. Introduction

Background

- **General-purpose models** for TS have been underexplored
- TS datasets
  - (1) Various **domains**
  - (2) Various **tasks**
- Current TS models: To transfer to new task/dataset, either require:
  - (1) Fine-tuning
  - (2) “Task/dataset-specific” modules

=> ***Lead to overfitting & hinder few/zero shot transfer***

2. UNITS

2-2. Introduction

Challenges of building TS foundation models

- (1) Inherent diverse & multi-domain TS
- (2) Diverging task specifications
- (3) Requirement for task-specific TS modules

2. UNITS

2-2. Introduction

Challenges of building TS foundation models

- (1) **Inherent diverse & multi-domain TS**
- (2) Diverging task specifications
- (3) Requirement for task-specific TS modules

- Unified models: Learn general knowledge by co-training on **DIVERSE** data
- TS: **Wide variability** in temporal dynamics across domains (= **heterogeneity**)

=> A unified model should capture "general temporal dynamics" that transfer to new downstream datasets (regardless of data representation)

2. UNITS

2-2. Introduction

Challenges of building TS foundation models

- (1) Inherent diverse & multi-domain TS
- (2) **Diverging task specifications**
- (3) Requirement for task-specific TS modules

- Tasks on TS have fundamentally **DIFFERENT** objectives
    - ex) FCST: predicting future values vs. CLS: discrete decision-making process
- **SAME** task across different datasets may require **DIFFERENT** specifications
    - ex) classification: 5-class vs. 10-class classification
    - ex) forecasting: 10 horizons vs 80 horizons future horizon

=> A unified model must be able to adapt to changing task specifications

2. UNITS

2-2. Introduction

Challenges of building TS foundation models

- (1) Inherent diverse & multi-domain TS
- (2) Diverging task specifications
- (3) **Requirement for task-specific TS modules**

Task-specific vs. Unified

- **Task-specific** modules: require **fine-tuning** these modules
- **Unified** models: employ **shared weights** across various tasks

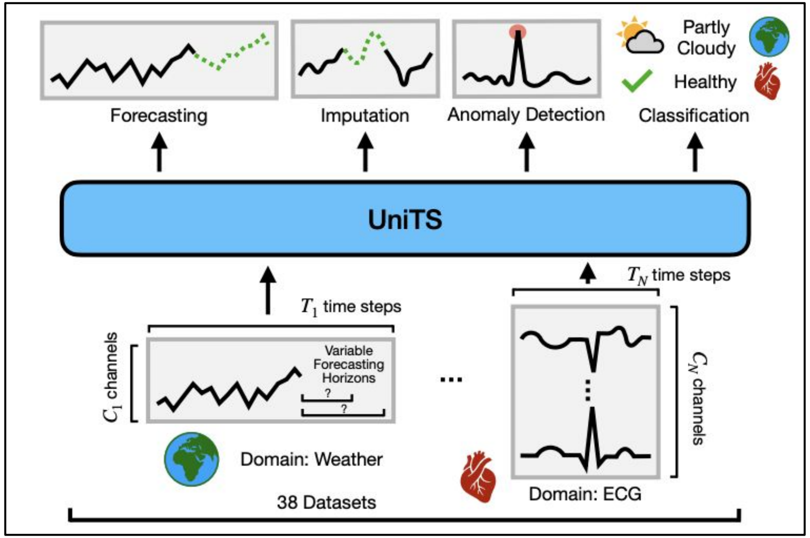
2. UNITS

2-2. Introduction

Ideal foundation TS model should ….

- (1) Inherent diverse & multi-domain TS
    - ***capture "general temporal dynamics" that transfer to new downstream datasets***
- (2) Diverging task specifications
    - ***adapt to changing task specifications***
- (3) Requirement for task-specific TS modules
    - ***employ shared weights across various tasks***

UniTS (Unified TS model)



2. UNITS

2-2. Introduction

UniTS (Unified TS model)

- Handles various tasks with shared parameters (task-specific modules X)
- Addresses the following challenges
  - (1) Universal task specification with prompting
  - (2) Data-domain agnostic network
  - (3) Fully shared weights

2. UNITS

2-2. Introduction

UniTS (Unified TS model)

- Handles various tasks with shared parameters (task-specific modules X)
- Addresses the following challenges
    - **(1) Universal task specification with prompting**
    - (2) Data-domain agnostic network
    - (3) Fully shared weights

Prompting: convert various tasks into a unified token representation

2. UNITS

2-2. Introduction

UniTS (Unified TS model)

- Handles various tasks with shared parameters (task-specific modules X)
- Addresses the following challenges
  - (1) Universal task specification with prompting
  - **(2) Data-domain agnostic network**
  - (3) Fully shared weights

- **Self-attention**: in both sequence & variable dimensions
- **Dynamic linear operator**: enables input with various lengths

2. UNITS

2-2. Introduction

UniTS (Unified TS model)

- Handles various tasks with shared parameters (task-specific modules X)
- Addresses the following challenges
  - (1) Universal task specification with prompting
  - (2) Data-domain agnostic network
  - **(3) Fully shared weights**

- **Shared weights** across tasks
- Unified pretraining scheme: **masked reconstruction**

2. UNITS

2-3. Problem Formulation

(1) Notation

[1] $\boxed{\textbf{Multi-domain datasets}}$ $D = \{D_i \mid i = 1, \ldots, n\}$,

- where each dataset D_i can have a **varying** number of TS
- can be of **varying** lengths & numbers of sensors/variables
- Each dataset: $D_i = (\mathcal{X}_i, \mathcal{Y}_i)$

[2] $\boxed{\text{Four tasks}}$

- forecasting
- classification
- anomaly detection
- imputation

2. UNITS

2-4. UniTS Model

(1) Overview

(2) Tokens

(3) Architecture

- Backbone
- Prediction Head

(4) Training

2. UNITS

2-4. UniTS Model

(1) Overview

Prompting-based model with a unified network

Three types of tokens

- (1) **Sample** token (= Time Series)
- (2) **Prompt** token (= Information of Dataset & Task)
- (3) **Task** tokens (= GEN + CLS token)

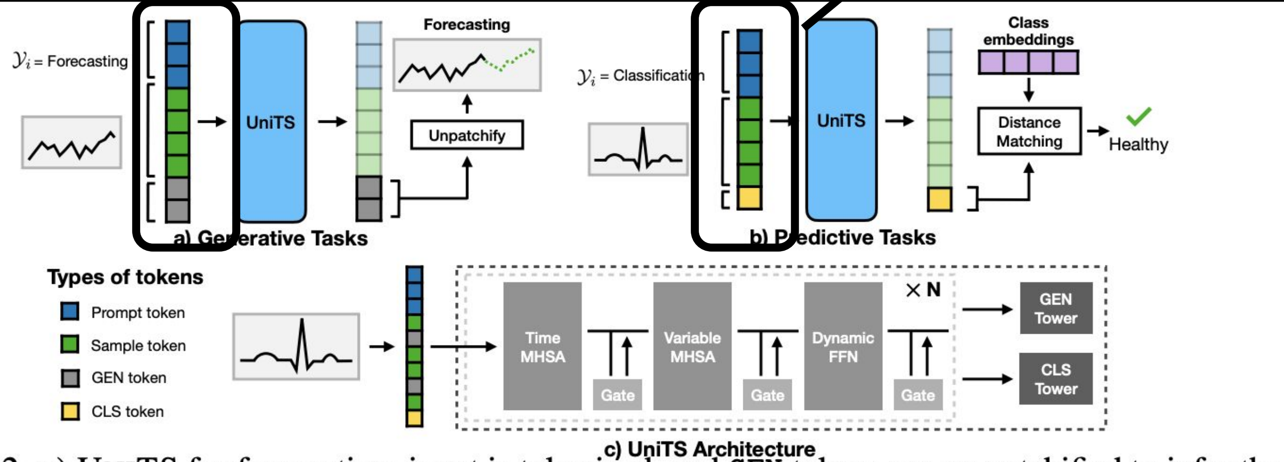


Figure 2: **a)** UNITS for forecasting; input is tokenized, and GEN tokens are un-patchified to infer the forecast horizon. **b)** UNITS for classification; a CLS token is used to represent class information and then compared to class tokens to get prediction class. **c)** Architecture of UNITS model.

Prompt

- 2개) FCST prompt + CLS prompt (task)
- 20개) Dataset1 prompt + .. Dataset10 prompt (dataset)
- 10개)
  - 전력 분야 forecasting (task+dataset)
  - 의료 분야 classification (task+dataset)
  - …
  - 공장 분야 forecasting (task+dataset)

Pretrained

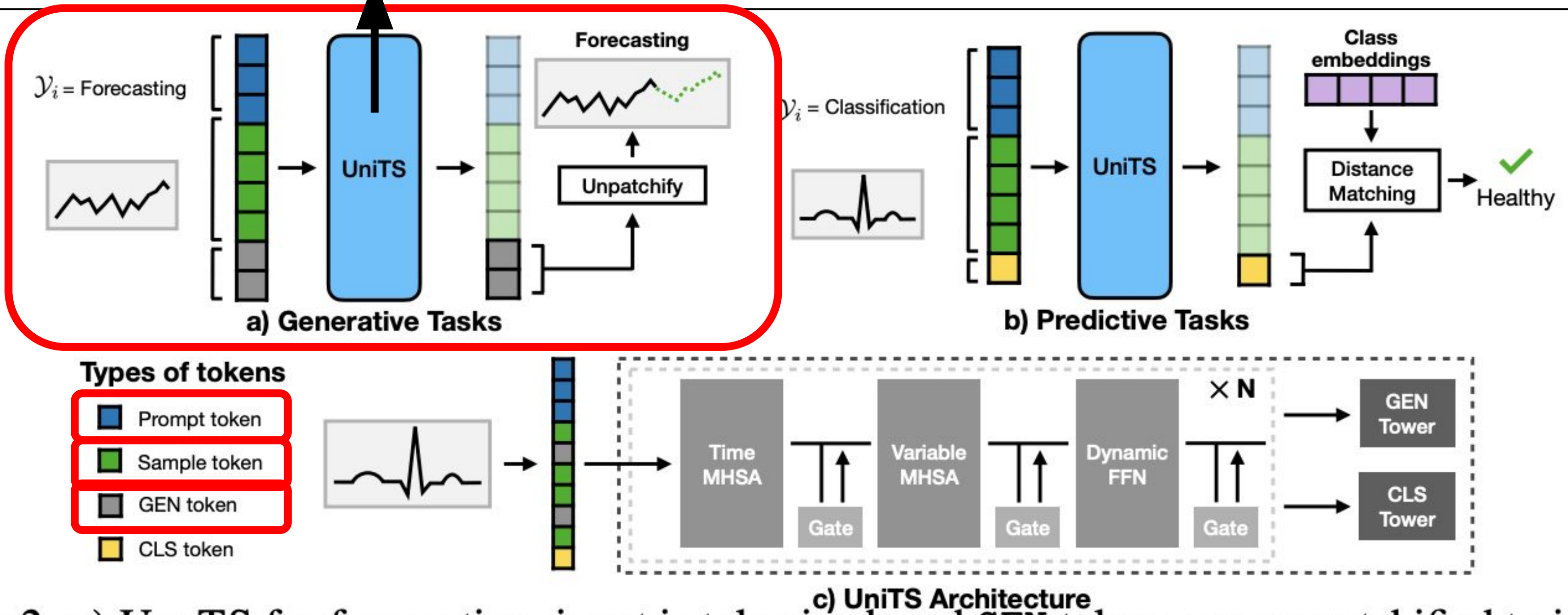


Figure 2: **a)** UNITS for forecasting; input is tokenized, and GEN tokens are un-patchified to infer the forecast horizon. **b)** UNITS for classification; a CLS token is used to represent class information and then compared to class tokens to get prediction class. **c)** Architecture of UNITS model.

(Downstream task) **Forecasting** task

Pretrained

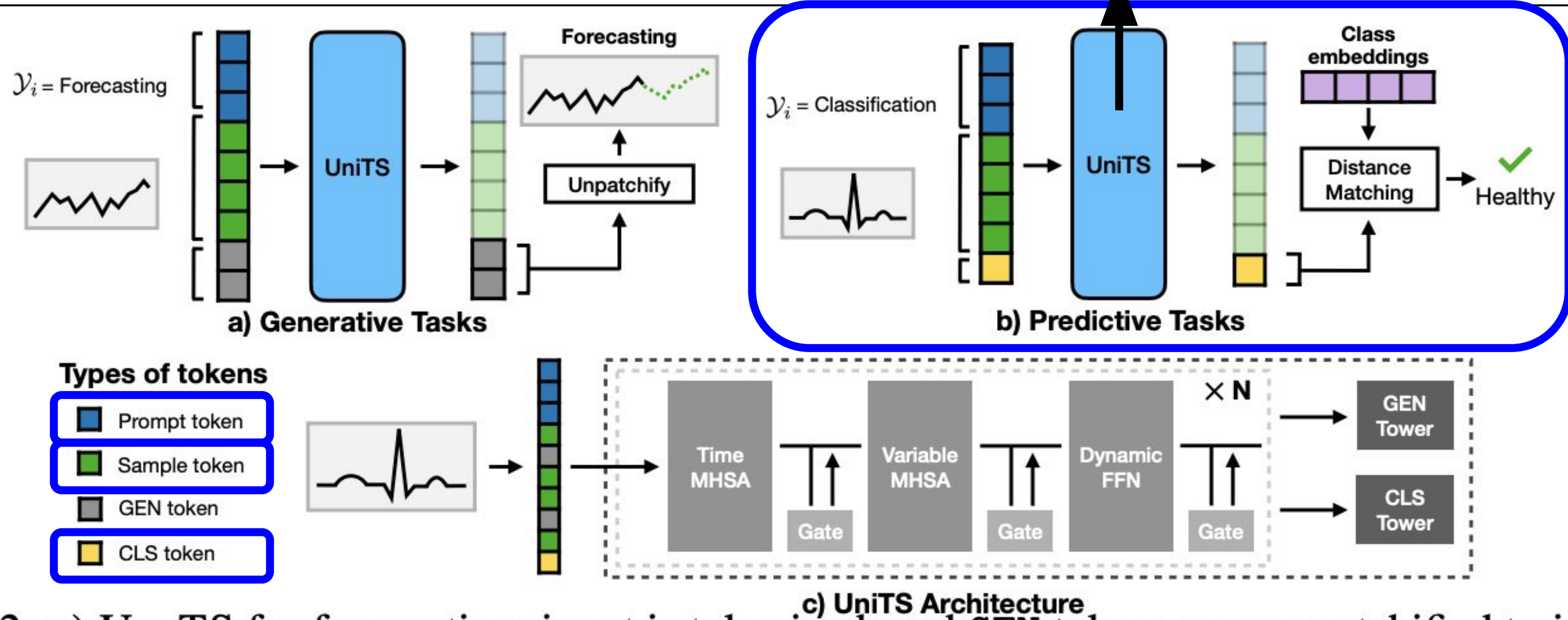


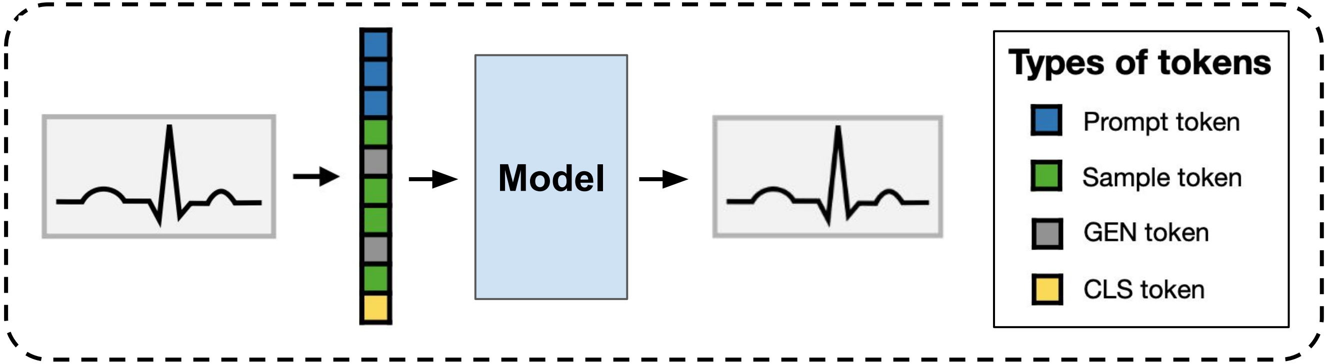
Figure 2: **a**) UNITS for forecasting; input is tokenized, and GEN tokens are un-patchified to infer the forecast horizon. **b**) UNITS for classification; a CLS token is used to represent class information and then compared to class tokens to get prediction class. **c**) Architecture of UNITS model.

(Downstream task) **Classification** task

2. UNITS

2-4. UniTS Model

(2) Tokens



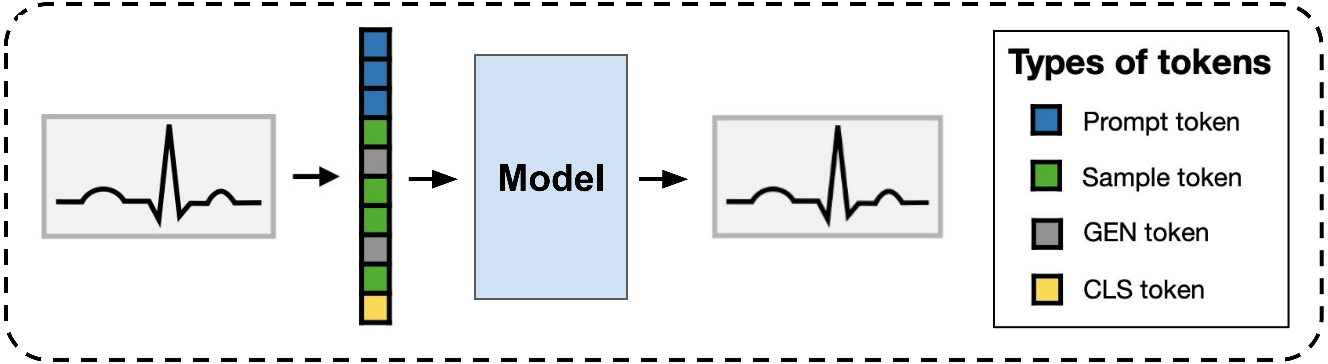
Three types of tokens

- (1) **Sample** token (= Time Series)
- (2) **Prompt** token (= Information of Dataset & Task)
- (3) **Task** tokens (= GEN + CLS token)

2. UNITS

2-4. UniTS Model

(2) Tokens



Three types of tokens

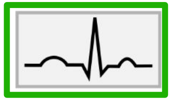
- (1) **Sample** token (= Time Series)
- (2) **Prompt** token (= Information of Dataset & Task)
- (3) **Task** tokens (= GEN + CLS token)

Length of TS

# of channels

# of sample patches

Patch dimension



$$\mathbf{x} \in \mathbb{R}^{\boxed{t}\times\boxed{v}} \Longrightarrow \mathbf{z}_{\mathbf{x}} \in \mathbb{R}^{\boxed{s}\times v\times\boxed{d}}$$

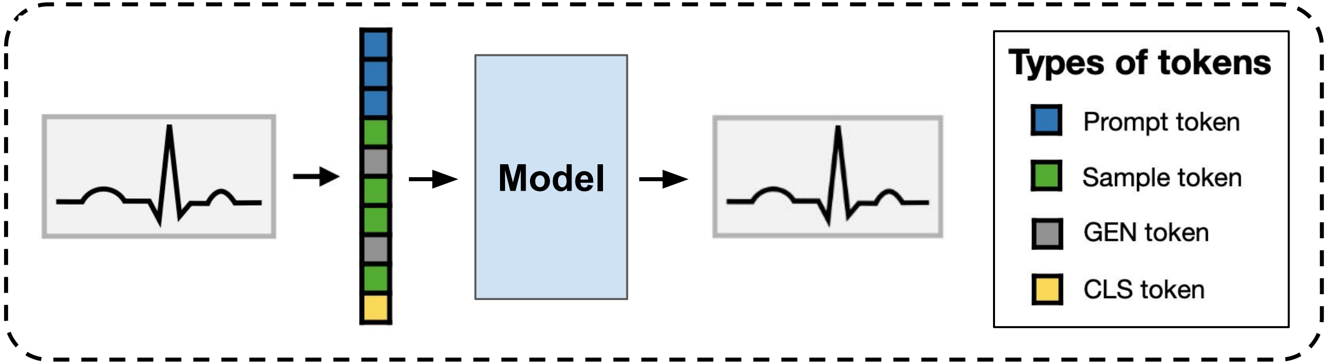
(+ Positional embedding)

Patchify & Embedding

2. UNITS

2-4. UniTS Model

(2) Tokens



Three types of tokens

- (1) **Sample** token (= Time Series)
- (2) **Prompt** token (= Information of Dataset & Task)
- (3) **Task** tokens (= GEN + CLS token)

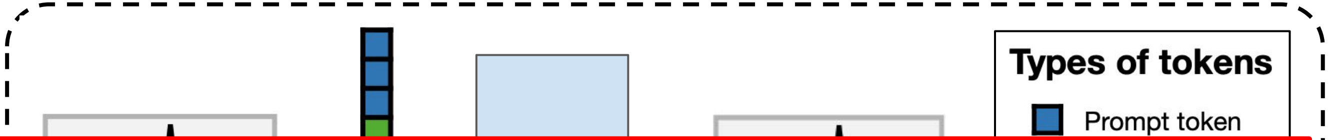
# of prompt tokens

$$\mathbf{z}_p \in \mathbb{R}^{\boxed{p}\times v\times d}$$

Learnable embeddings

Each dataset has its own set of prompt tokens

2. UNITS



D.1 Model Details

By default, in a multi-task setting, the UNITS network comprises three UNITS blocks, one GEN tower, and one CLS tower. For each data source, the prompt tokens and task tokens are defined. Forecasting tasks on the same data source but with different forecast lengths share the same prompt and GEN token. For zero-shot learning on new datasets, we use a shared prompt and GEN token across all data sources to facilitate zero-shot learning. Tokens are trained to achieve their functions. The number of embedding dimensions, d, is set to 64 for UNITS-*SUP* and 128 for UNITS-*PMT*. All blocks in UNITS maintain the same feature shape, following the Transformer architecture.

$$\mathbf{z}_p \in \mathbb{R}^{\boxed{p} \times v \times d}$$

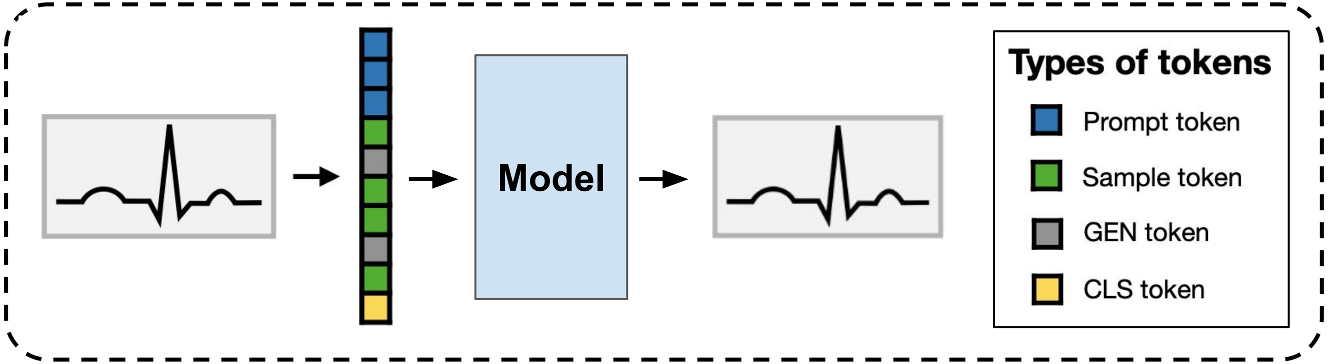
Learnable embeddings

Each dataset has its own set of prompt tokens

2. UNITS

2-4. UniTS Model

(2) Tokens



Three types of tokens

- (1) **Sample** token (= Time Series)
- (2) **Prompt** token (= Information of Dataset & Task)
- (3) **Task** tokens (= GEN + CLS token)

Two types of task tokens:

- GEN: generation token (for forecasting, imputation, anomaly detection)
- CLS: classification token

replicated f-times based on desired forecasting length to get $\hat{\mathbf{z}}_m \in \mathbb{R}^{f \times v \times d}$.

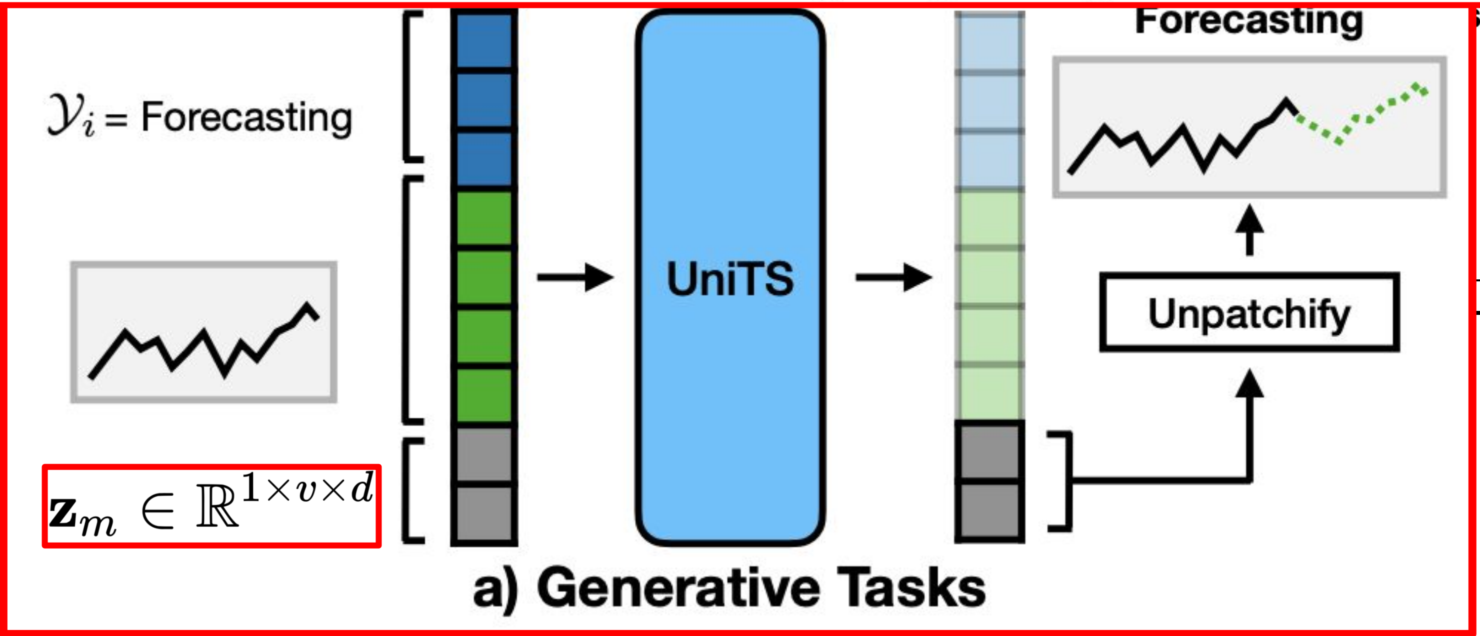
2. UNITS

2-4. UniTS Mo

(2) Tokens

Three types of to

- (1) **Sample**
- (2) **Prompt**
- (3) **Task to**



Two types of task tokens:

- GEN: **generation token** (for forecasting, imputation, anomaly detection)
- CLS: classification token

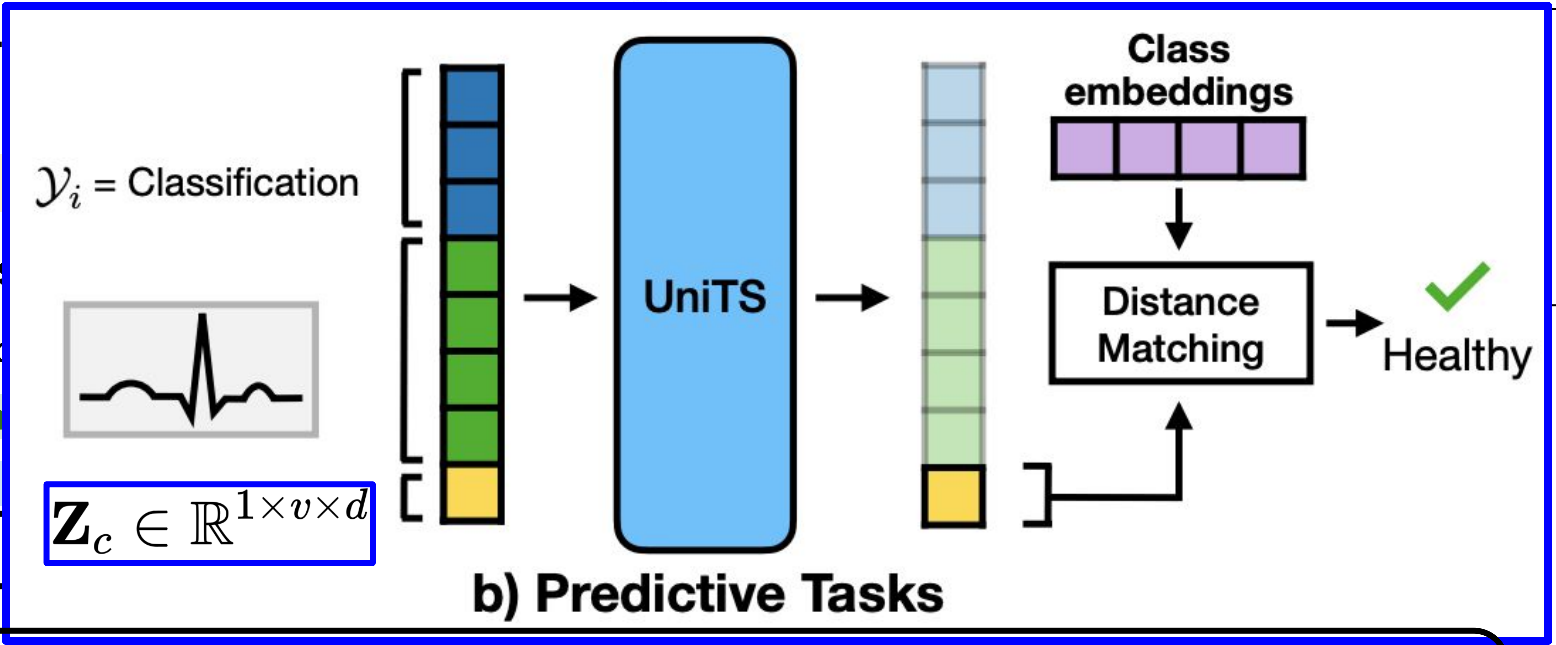
2. UNIT

2-4. UniTS

(2) Token

Three types o

- (1) **San**
- (2) **Pro**
- (3) **Tas**



Two types of task tokens:

- GEN: generation token (for forecasting, imputation, anomaly detection)
- CLS: **classification** token

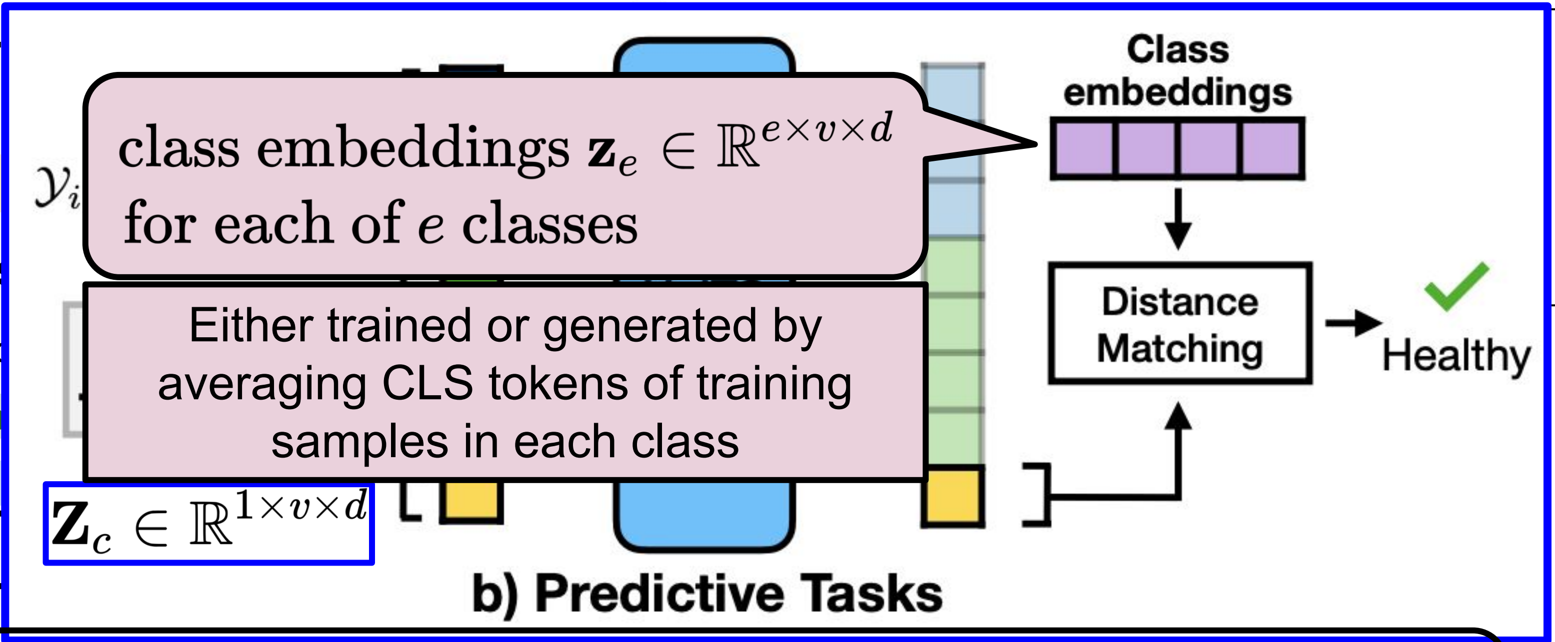
2. UNIT

2-4. UniTS

(2) Tokens

Three types

- (1) San
- (2) Pro
- (3) Tas



Two types of task tokens:

- GEN: generation token (for forecasting, imputation, anomaly detection)
- CLS: **classification** token

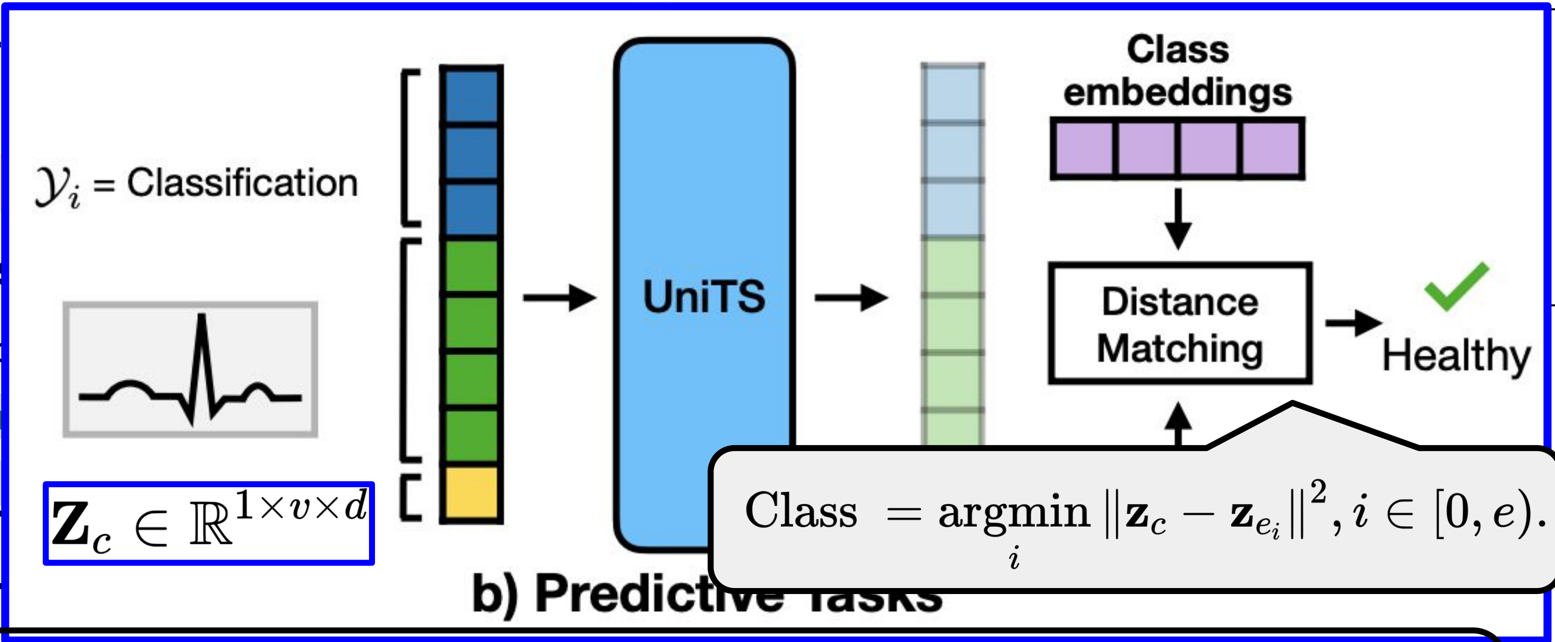
2. UNIT

2-4. UniTS

(2) Token

Three types o

- (1) San
- (2) Pro
- (3) Tas

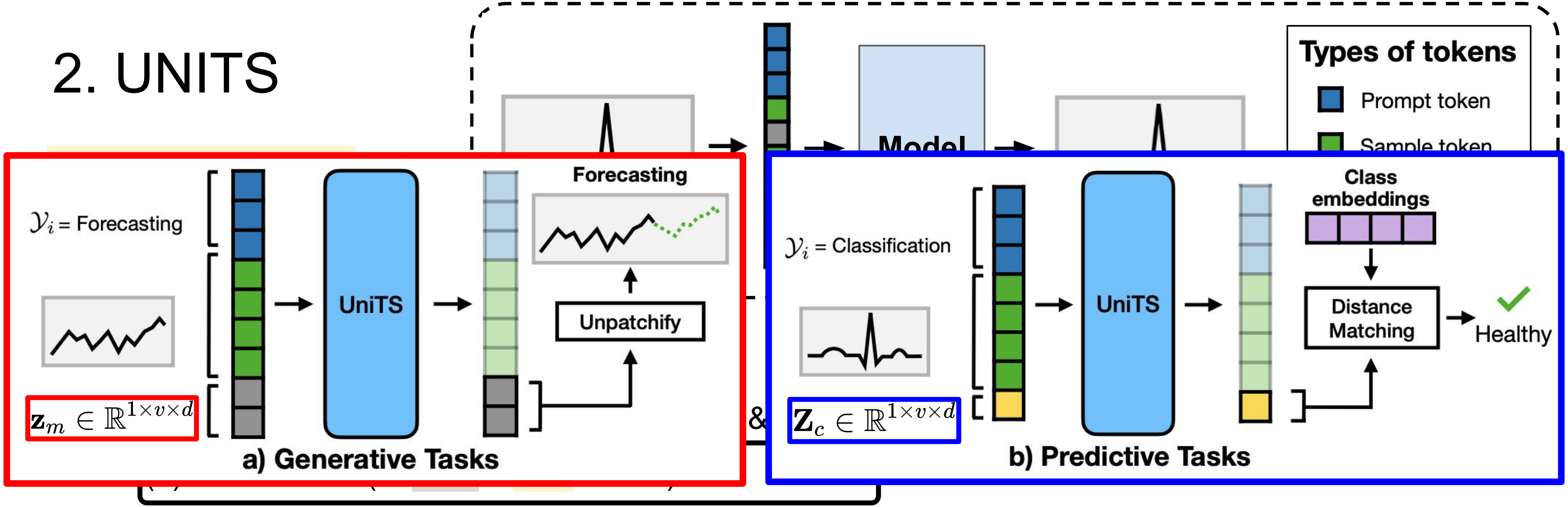


$$\text{Class} = \underset{i}{\operatorname{argmin}} \left\|\mathbf{z}_c - \mathbf{z}_{e_i}\right\|^2, i \in [0, e).$$

Two types of task tokens:

- GEN: generation token (for forecasting, imputation, anomaly detection)
- CLS: **classification** token

2. UNITS



Both (GEN & CLS) are concatenated along the time dimension

- with the prompt and sample tokens

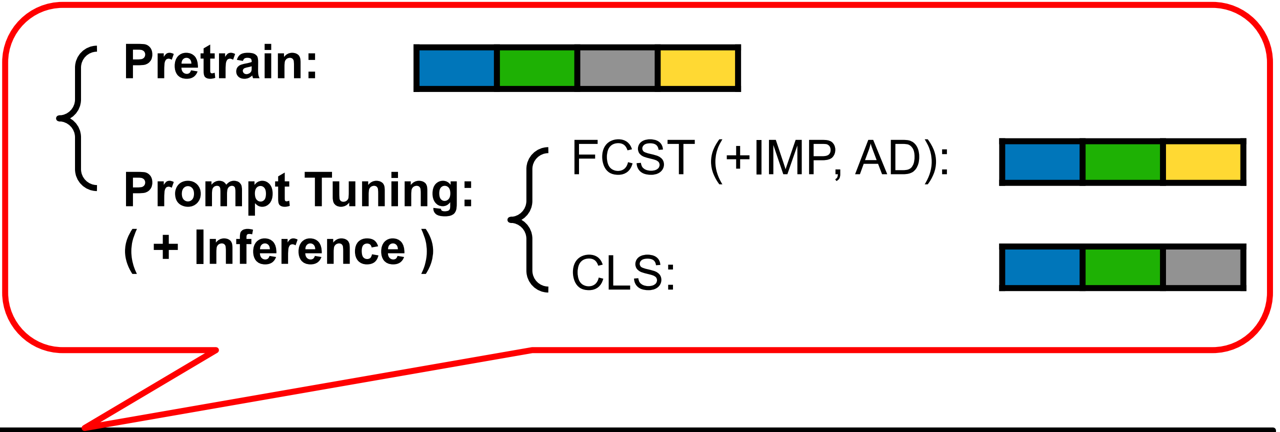
$$\mathbf{z}_{\text{Fore}} = \mathbf{CA}(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}}, \hat{\mathbf{z}}_m) \in \mathbb{R}^{(p+s+f)\times v\times d}$$

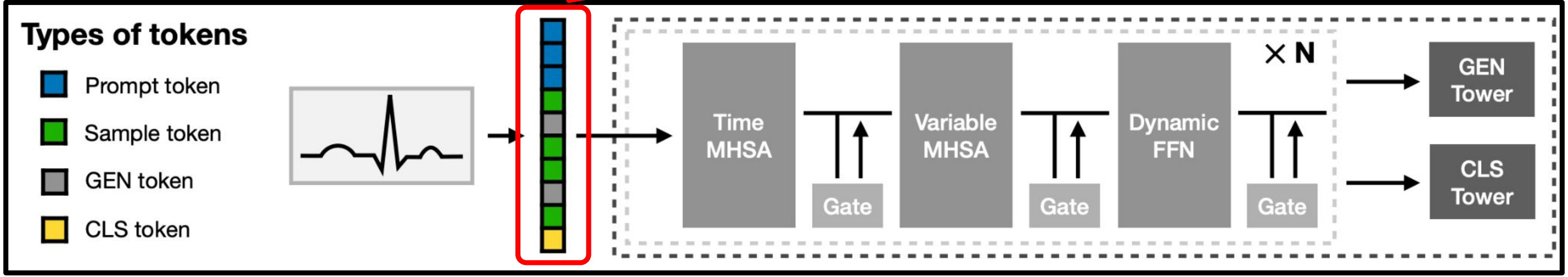
$$\mathbf{z}_{\text{Pred}} = \mathbf{CA}(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}}, \mathbf{z}_c) \in \mathbb{R}^{(p+s+1)\times v\times d}$$

2. UNITS

2-4. UniTS Model

(2) Tokens

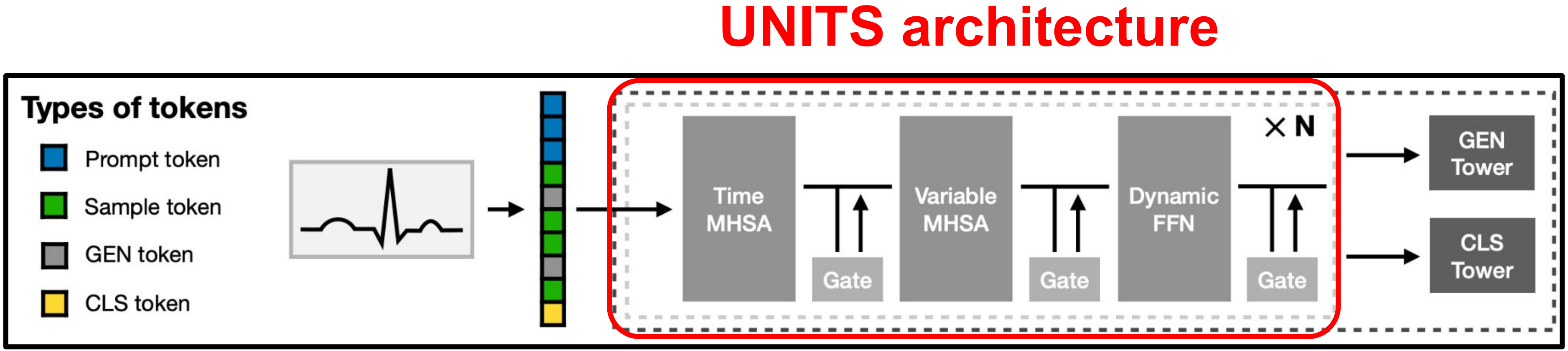




2. UNITS

2-4. UniTS Model

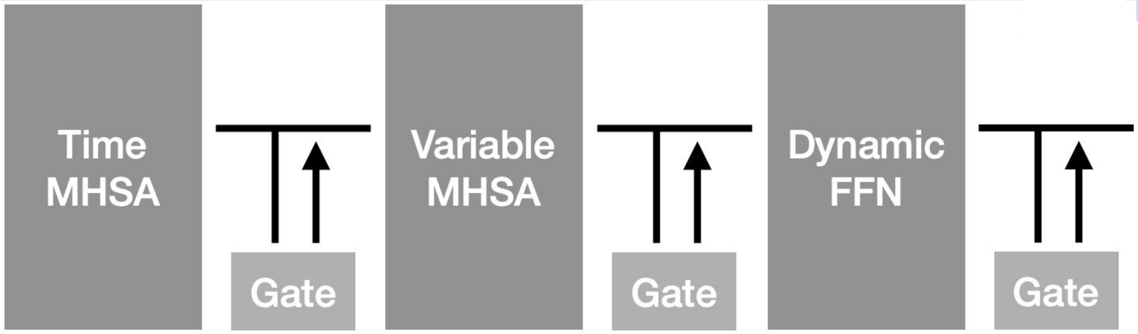
(3) Architecture - Backbone



=> Three main components!

2. UNITS

2-4. UniTS Model



(3) Architecture - Backbone

a) Multi Head Self-attention (MHSA)

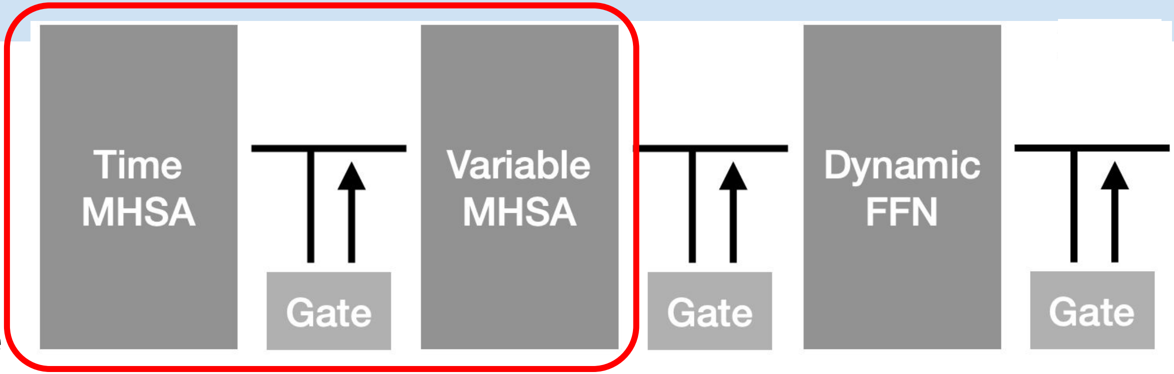
- a-1) **Time MHSA**
- a-2) **Variable MHSA**

b) **Dynamic FFN**

2. UNITS

2-4. UniTS Model

(3) Architecture - Backbone



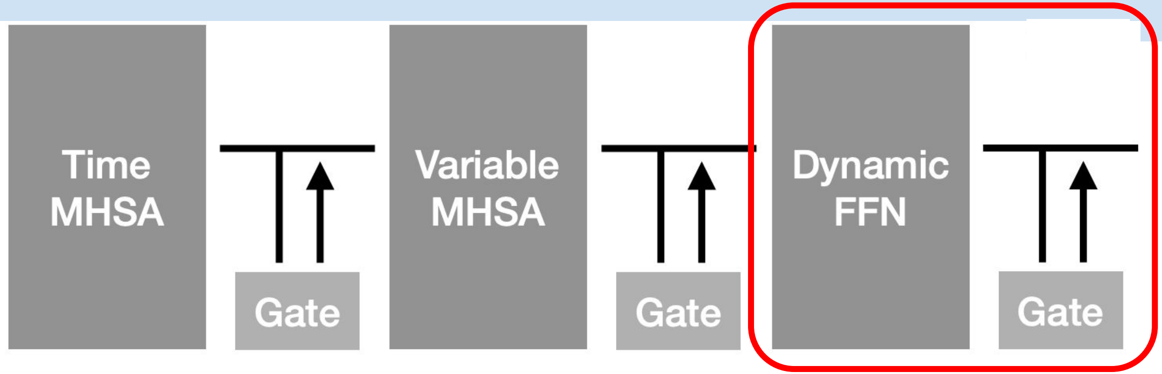
a) Multi Head Self-attention (MHSA)

- Previous methods) only either **Time** or **Variable** dimension
- **Time MHSA + Variable MHSA**
  - effectively handle TS samples with various **L** & **C**

2. UNITS

2-4. UniTS Model

(3) Architecture - Backbone



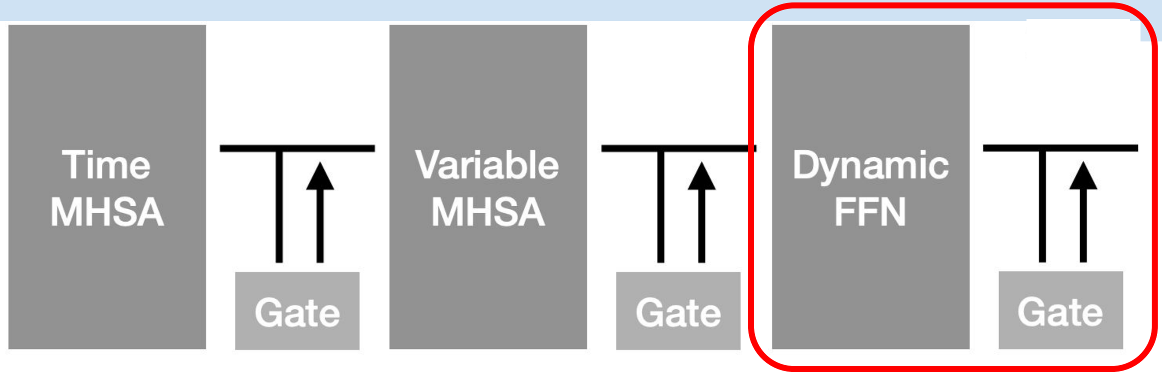
b) Dynamic FFN

- Incorporate a dynamic operator (Dylinear) into FFN
- Dylinear = **Weight interpolation** scheme
  - Enables the FFN to capture dependencies between tokens
  - Accommodate varying time lengths.
  - Different variables in a sample share one DyLinear layer

2. UNITS

2-4. UniTS Model

(3) Architecture - Backbone



b) Dynamic FFN

- Dylinear = Weight interpolation scheme

Interpolated weight

learnable weights $\mathbf{w} \in \mathbb{R}^{w_{\text{in}} \times w_{\text{out}}}$

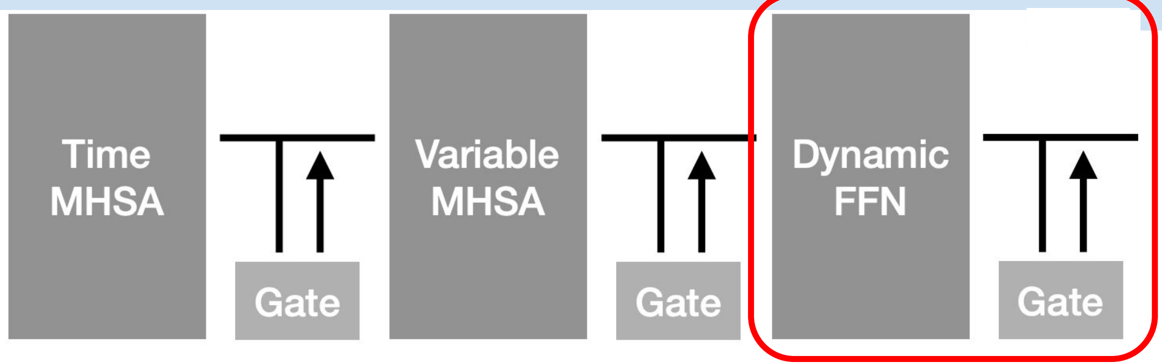
$$\boxed{\mathbf{W}_{\text{Interp}} = \text{Interp}(\boxed{\mathbf{w}}) \in \mathbb{R}^{l_{\text{in}} \times l_{\text{out}}}}$$

where Interp is a bi-linear interpolation to resize $\mathbf{w}$ from shape $w_{\text{in}} \times w_{\text{out}}$ to $l_{\text{in}} \times l_{\text{out}}$

2. UNITS

2-4. UniTS Model

(3) Architecture - Backbone



b) Dynamic FFN

- Dylinear = Weight interpolation scheme

Input

tokens $\mathbf{z}_t \in \mathbb{R}^{l_{\text{in}} \times d}$

$$\operatorname{DyLinear}\left(\mathbf{z}_t, \mathbf{w}\right)=\boxed{\mathbf{W}_{\text{Interp}}}\,\boxed{\mathbf{z}_t} \in \mathbb{R}^{l_{\text{out}} \times d}$$

learnable weights $\mathbf{w} \in \mathbb{R}^{w_{\text{in}} \times w_{\text{out}}}$

Interpolated weight

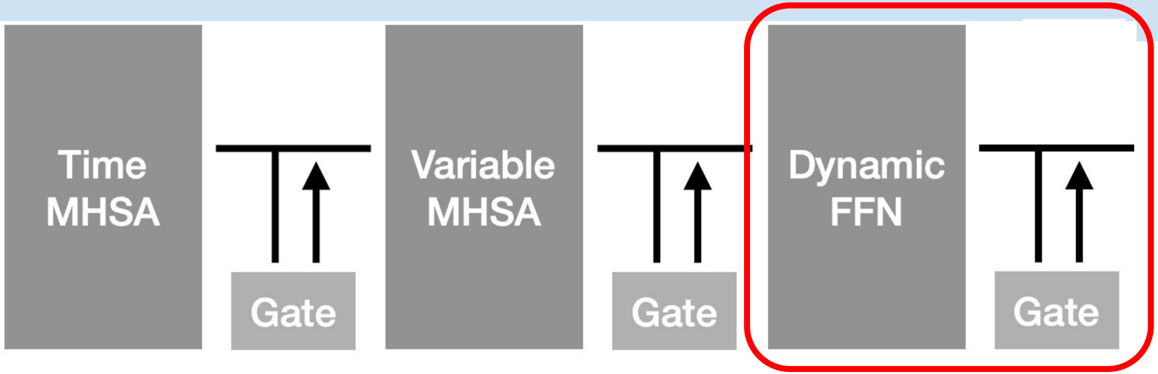
$$\boxed{\mathbf{W}_{\text{Interp}}=\operatorname{Interp}(\boxed{\mathbf{w}}) \in \mathbb{R}^{l_{\text{in}} \times l_{\text{out}}}}$$

where Interp is a bi-linear interpolation to resize $\mathbf{w}$ from shape $w_{\text{in}} \times w_{\text{out}}$ to $l_{\text{in}} \times l_{\text{out}}$

2. UNITS

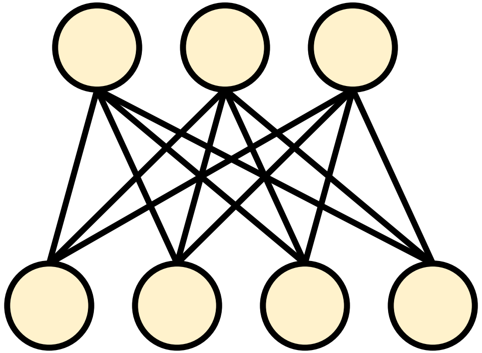
2-4. UniTS Model

(3) Architecture - Backbone



b) Dynamic FFN

- Dylinear = Weight interpolation scheme



(default shape)

learnable weights $\mathbf{w} \in \mathbb{R}^{w_{\text{in}} \times w_{\text{out}}}$

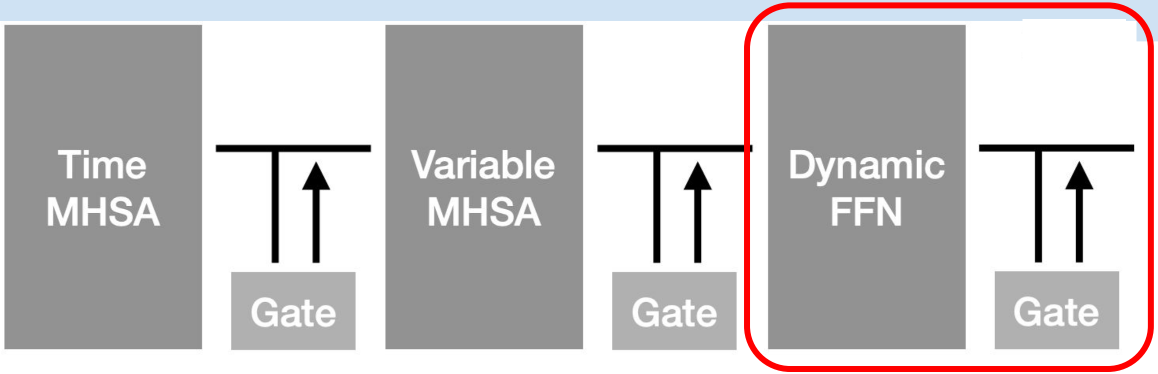
$$w_{\text{in}} = 4$$

$$w_{\text{out}} = 3$$

2. UNITS

2-4. UniTS Model

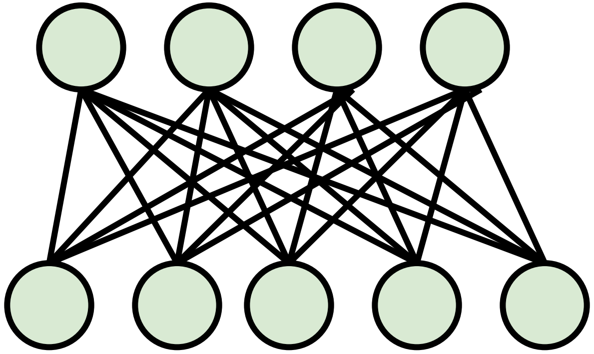
(3) Architecture - Backbone



b) Dynamic FFN

```
dynamic_weights = F.interpolate(dynamic_weights.unsqueeze(0).unsqueeze(0), size=(
    out_features, in_features-self.fixed_in), mode='bilinear', align_corners=False).squeeze(0).squeeze(0)
```

- Dylinear = Weight interpolation scheme



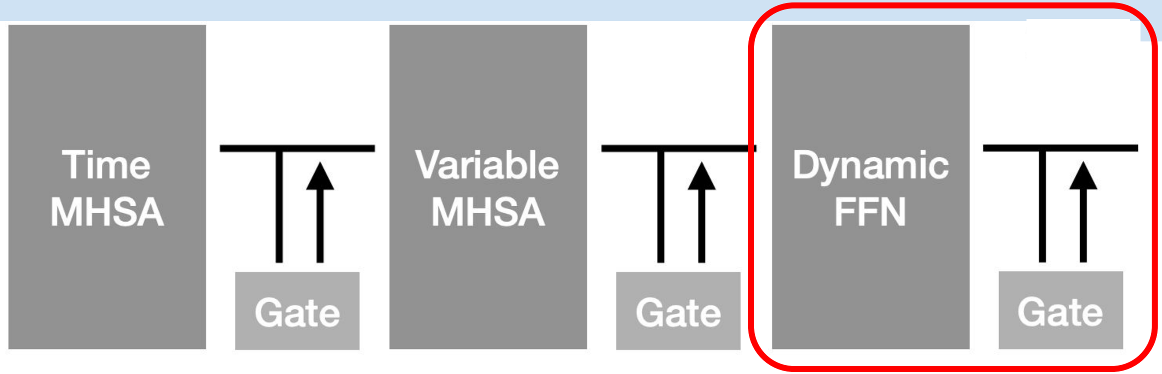
Interpolated!
Enables variable input & output dimension (length)

$$\mathbf{W}_{\text{Interp}} = \text{Interp}(\mathbf{w}) \in \mathbb{R}^{l_{\text{in}} \times l_{\text{out}}}$$
$$l_{\text{in}} = 5$$
$$l_{\text{out}} = 4$$

2. UNITS

2-4. UniTS Model

(3) Architecture - Backbone



b) Dynamic FFN

- Intuitive in TS domain!
  - TS samples, even with different sampling rates, exhibit similar dynamic patterns.
- Substantial performance improvements in generative tasks (Table 19)

Table 19: Ablation on the Dynamic FFN layer in UNITS network.

	$Acc_{Avg}\uparrow$	$MSE_{Avg}\downarrow$	$MAE_{Avg}\downarrow$
UNITS-*SUP*	**81.6**	**0.439**	**0.381**
Dynamic FFN → FFN	81.3	0.462	0.394
Without Dynamic FFN	80.8	0.465	0.396

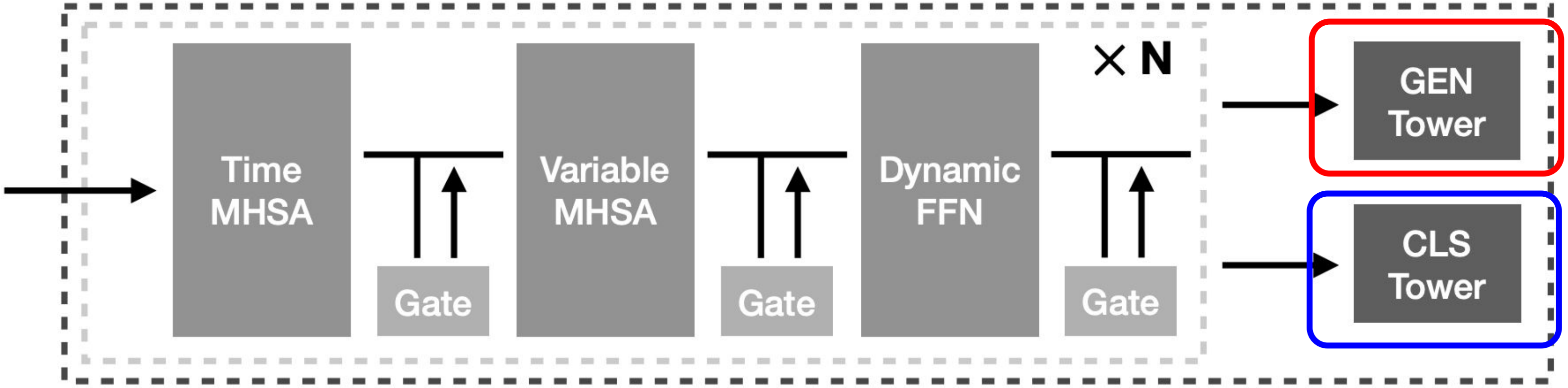
2. UNITS

2-4. UniTS Model

(3) Architecture - Prediction

a) GEN (generation) tower $\boxed{H_{\mathrm{GEN}}}$

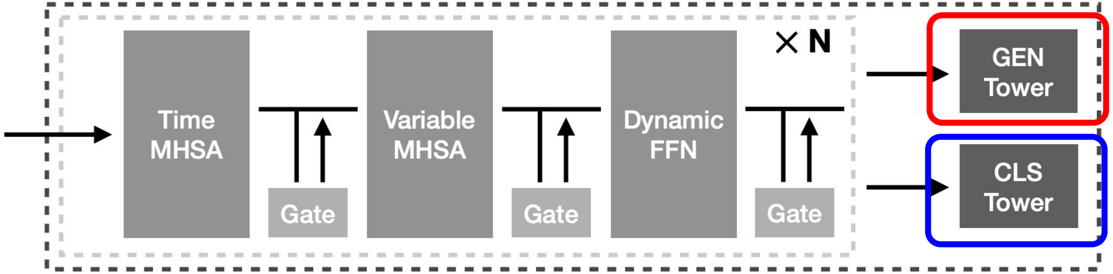
b) CLS (classification) tower $\boxed{H_{\mathrm{CLS}}}$



2. UNITS

2-4. UniTS Model

(3) Architecture - Prediction

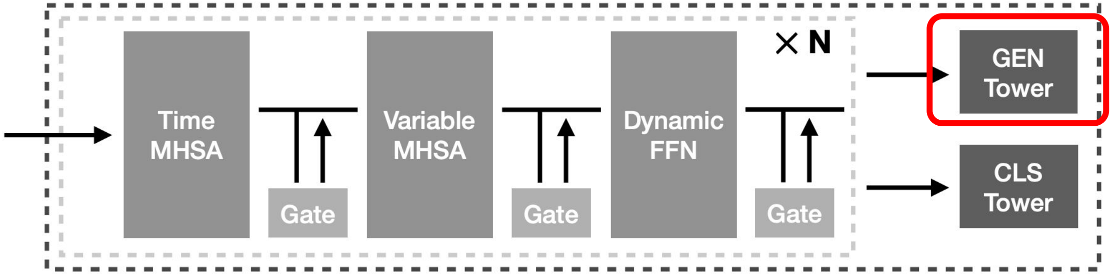


- Shared towers for all datasets & tasks! … ***task/dataset-specific (X)***
- Unified & efficient model architecture
- Pre-training & Prompt-tuning
  - Pre-training) train both **GEN** + **CLS** tower
  - Prompt-Tuning)
    - **Generative** task (FCST,IMP,AD): use only **GEN** tower
    - **Classification** task: use only **CLS** tower

2. UNITS

2-4. UniTS Model

(3) Architecture - Prediction



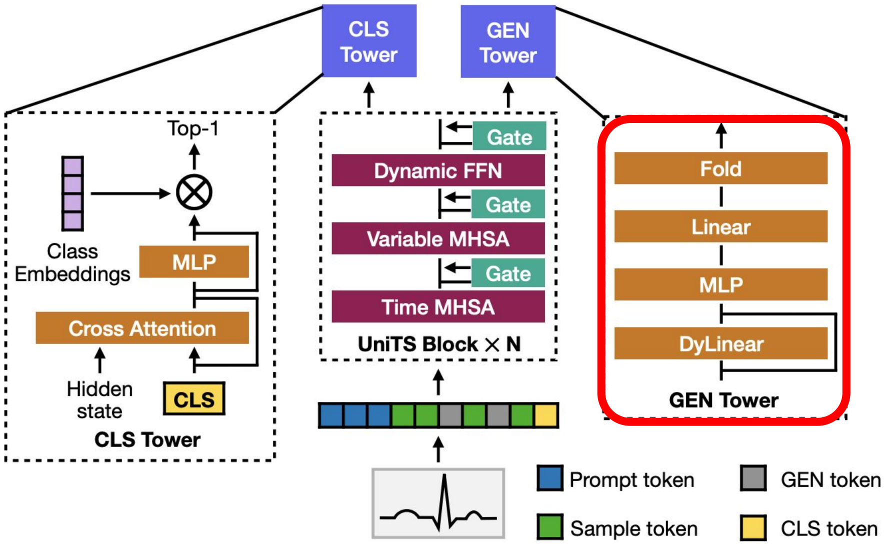
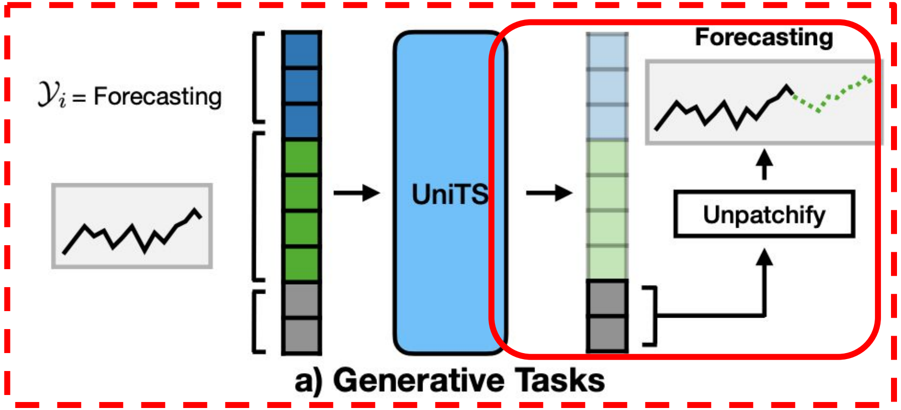


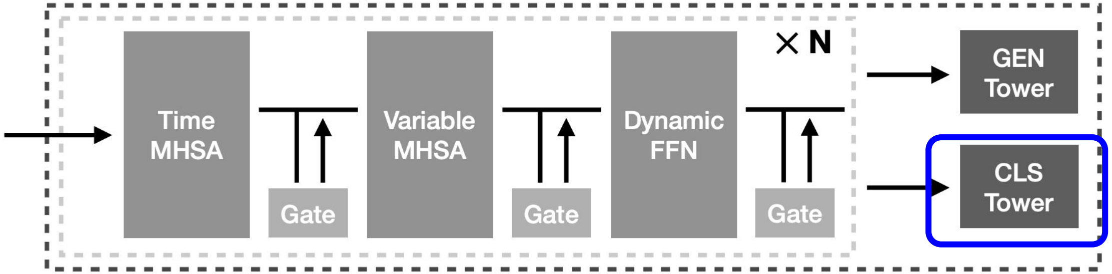
Figure 4: The network architecture of UNITS. Shared GEN tower and CLS tower transform task tokens to the prediction results of generative and predictive tasks.



2. UNITS

2-4. UniTS Model

(3) Architecture - Prediction



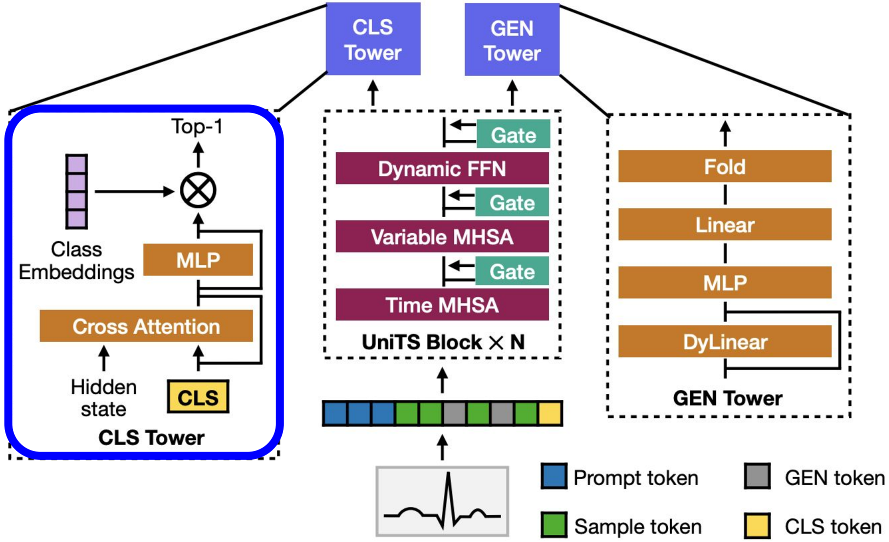
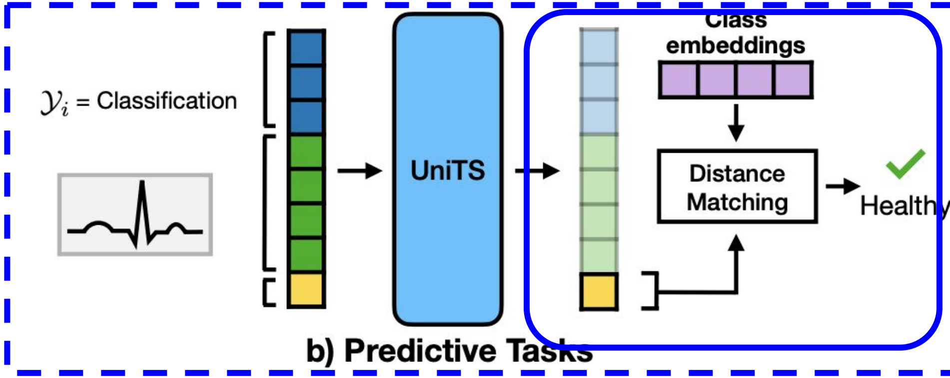


Figure 4: The network architecture of UNITS. Shared GEN tower and CLS tower transform task tokens to the prediction results of generative and predictive tasks.



2. UNITS

2-4. UniTS Model

(4) Training

a) Goal of pre-training

- **# 1**) Learn **general features**
    - Applicable to both **generative & predictive** tasks
- **# 2**) Efficiently **adapt** to downstream tasks
    - Via **prompt learning**

2. UNITS

2-4. UniTS Model

(4) Training

b) Comparison with previous works

- Previous)
    - (1) **Either** generative or predictive
    - (2) Pretrain **only** the model **backbone**
- UniTS)
    - (1) **Both** generative and predictive
    - (2) Pretrain **all** model (**backbone + towers**) => ***Enables zero-shot over frozen model!***

2. UNITS

2-4. UniTS Model

(4) Training

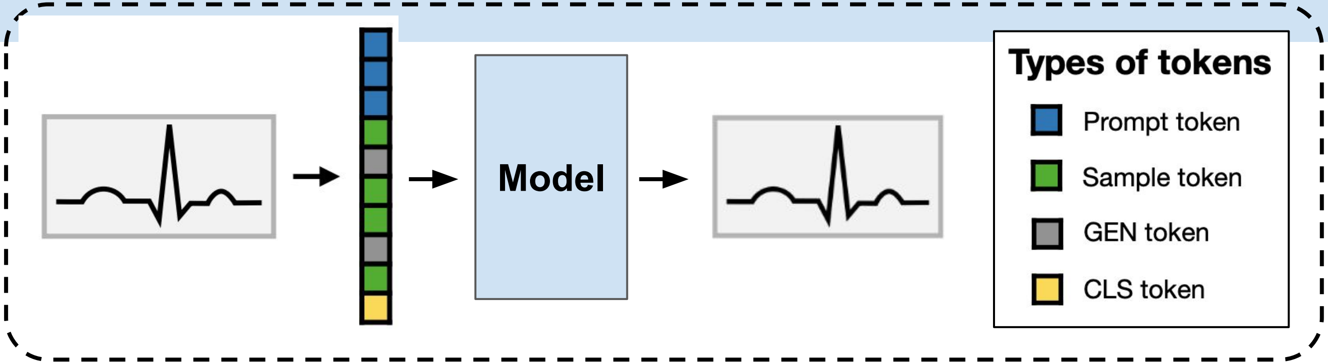
c) Pretraining task: Unified masked reconstruction pretraining

- **Unified**) Same pretraining tasks all datasets (FCST & CLS)
- **Masked reconstruction**) Loss of all (masked & unmasked) parts
- Leverages the semantics of both **prompt & CLS tokens**

2. UNITS

2-4. UniTS Model

(4) Training



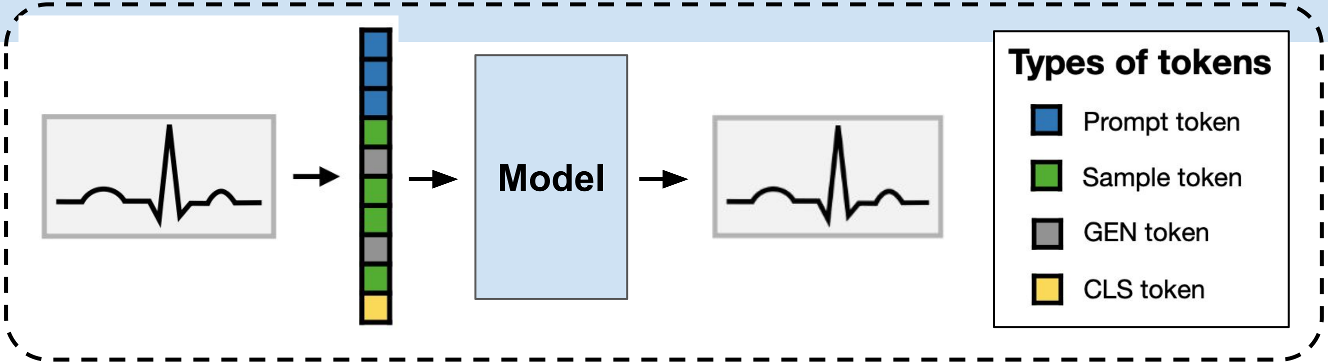
c) Pretraining task: Unified masked reconstruction pretraining

$$L_{\text{pretrain}} = L_{\text{MSE}}(H_{\text{GEN}}(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}}), \mathbf{x}) + L_{\text{MSE}}(H_{\text{GEN}}(H_{\text{CLS}}(\mathbf{z}_{\text{Pred}}), \mathbf{z}_{\mathbf{x}}), \mathbf{x})$$

2. UNITS

2-4. UniTS Model

(4) Training



c) Pretraining task: Unified masked reconstruction pretraining

$$L_{\text{pretrain}} = L_{\text{MSE}}\left(\boxed{H_{\text{GEN}}(\boxed{\mathbf{z}_p}, \boxed{\mathbf{z}_{\mathbf{x}}})}, \mathbf{x}\right) + L_{\text{MSE}}\left(\boxed{H_{\text{GEN}}(H_{\text{CLS}}(\mathbf{z}_{\text{Pred}}), \boxed{\mathbf{z}_{\mathbf{x}}})}, \mathbf{x}\right)$$

$$\mathbf{z}_{\text{Pred}} = \mathbf{CA}(\boxed{\mathbf{z}_p}, \boxed{\mathbf{z}_{\mathbf{x}}}, \boxed{\mathbf{z}_c}) \in \mathbb{R}^{(p+s+1)\times v\times d}$$

Sum of two reconstruction losses, focusing on

- (1) **Generative task (GEN tower)**
- (2) **Classification task (CLS tower)**

2. UNITS

2-4. UniTS Model

(4) Training

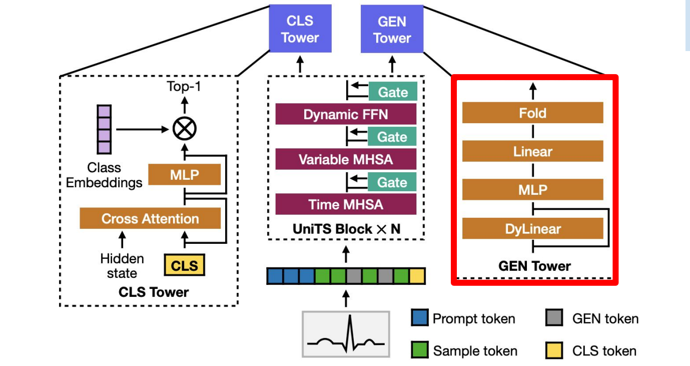


Figure 4: The network architecture of UNITS. Shared GEN tower and CLS tower transform task tokens to the prediction results of generative and predictive tasks.

c) Pretraining task: Unified masked reconstruction pretraining

$$L_{\text{pretrain}} = L_{\text{MSE}}(\boxed{H_{\text{GEN}}(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}})}, \mathbf{x}) + L_{\text{MSE}}(H_{\text{GEN}}(H_{\text{CLS}}(\mathbf{z}_{\text{Pred}}), \mathbf{z}_{\mathbf{x}}), \mathbf{x})$$

prompt token $\mathbf{z}_p$

sample token $\mathbf{z}_{\mathbf{x}}$

$\boxed{\text{GEN tower } H_{\text{GEN}}}$

2. UNITS

2-4. UniTS Model

(4) Training

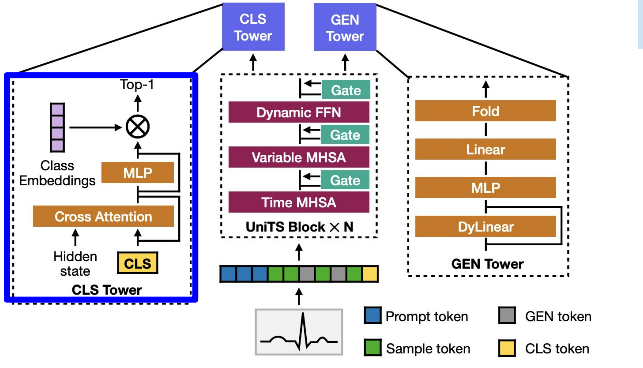


Figure 4: The network architecture of UniTS. Shared GEN tower and CLS tower transform task tokens to the prediction results of generative and predictive tasks.

c) Pretraining task: **Unified masked reconstruction** pretraining

$$L_{\text{pretrain}} = L_{\text{MSE}}\left(H_{\text{GEN}}\left(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}}\right), \mathbf{x}\right) + L_{\text{MSE}}\left(\boxed{H_{\text{GEN}}\left(H_{\text{CLS}}\left(\mathbf{z}_{\text{Pred}}\right), \mathbf{z}_{\mathbf{x}}\right)}, \mathbf{x}\right)$$

- To leverage the semantics of CLS token
- To train the CLS tower (for predictive tasks)

$$\boxed{\hat{\mathbf{z}}_{\text{Pred}} = H_{\text{CLS}}\left(\mathbf{z}_{\text{Pred}}\right)}$$

(prompt + TS +cls)

$$\mathbf{z}_{\text{Pred}} = \mathbf{CA}\left(\mathbf{z}_p, \mathbf{z}_{\mathbf{x}}, \mathbf{z}_c\right) \in \mathbb{R}^{\boxed{(p+s+1)} \times v \times d}$$

2. UNITS

2-4. UniTS Model

(4) Training

d) Two model versions

- Version 1) PMT (Prompt-tuning)
    - Pre-trained UniTS (via reconstruction task)
    - Freeze weight + Tune (prompt & task) tokens
- Version 2) SUP (Supervised)
    - Standard multi-task supervised learning
    - Single model is trained from scratch to perform many tasks.

2. UNITS

2-4. UniTS Model

(4) Training

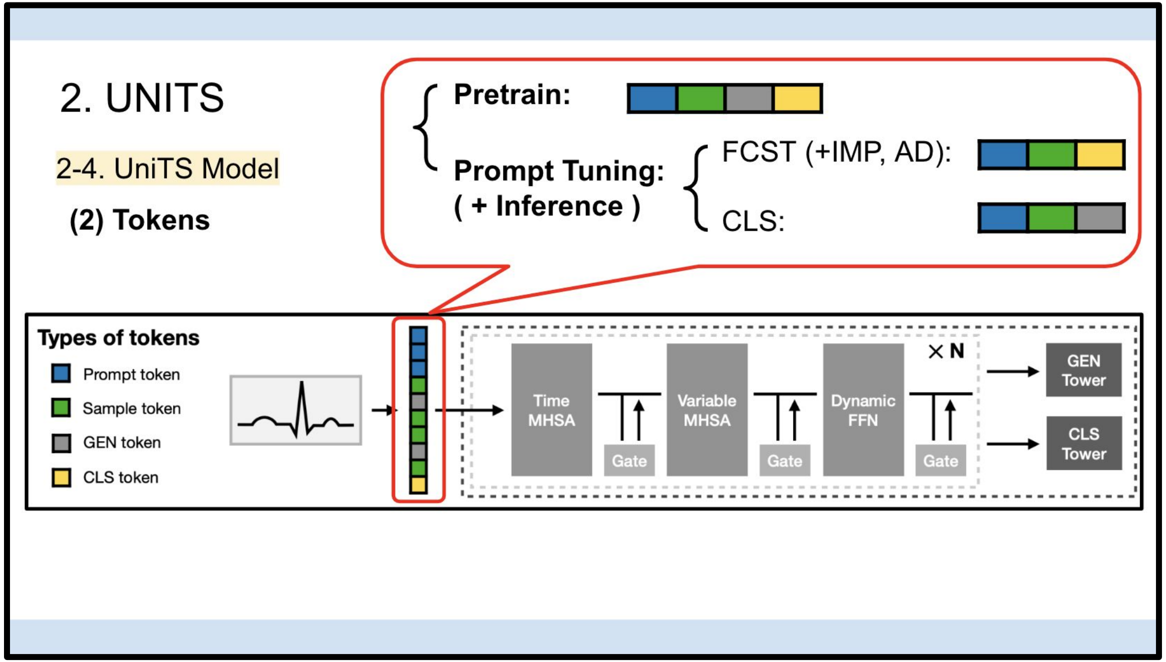
e) Summary

Unified pretraining strategy

that pretrains….

- **(1) Tokens**
- (2) Backbone network
- (3) GEN/CLS towers

for both generative and predictive abilities.



2. UNITS

2-4. UniTS Model

(5) Experiments

a) Datasets & Tasks

b) Single-task Learning -> skip

c) Multi-task Learning

d) Forecasting new horizons

e) Few-shot Learning

2. UNITS

2-4. UniTS Model

(5) Experiments

a) Datasets & Tasks

- **38 datasets** (= actually, **25** datasets)
  - **Forecasting**: 20 datasets (= actually, **7** datasets)
  - **Classification**: 18 datasets
- **38 Tasks**
- For few-shot learning, additional **6 datasets (for IMP)** & **5 datasets (for AD)**

Different prediction horizon with same datasets

MULTI-TASK FORECAST
NN5_{P112}
ECL_{P96}
ECL_{P192}
ECL_{P336}
ECL_{P720}
ETTH1_{P96}
ETTH1_{P192}
ETTH1_{P336}
ETTH1_{P720}

2. UNITS

2-4. UniTS Model

(5) Experiments

Table 2: Multi-task benchmarking across 20 forecasting tasks and 18 classification tasks. Both UNITS-*SUP* and UNITS-*PMT* process all 38 tasks using a single model. GPT4TS reprograms a pre-trained LLM (GPT-2) to time series and has dataset/task-specific modules. “$_P$” is forecasting length. “Class./Num.” denotes the “number of classes in each task”/“number of datasets”.

c) Multi-task Learning

MULTI-TASK FORECAST	UNITS-*SUP* MSE↓	UNITS-*SUP* MAE↓	UNITS-*PMT* MSE↓	UNITS-*PMT* MAE↓	ITRANS. MSE↓	ITRANS. MAE↓	TIMESNET MSE↓	TIMESNET MAE↓	PATCHTST MSE↓	PATCHTST MAE↓	PYRAFORMER MSE↓	PYRAFORMER MAE↓	AUTOFORMER MSE↓	AUTOFORMER MAE↓	GPT4TS MSE↓	GPT4TS MAE↓
NN5$_{P112}$	**.611**	.549	.622	.546	.623	.554	.629	**.541**	.634	.568	1.07	.791	1.23	.903	.623	.545
ECL$_{P96}$	.167	.271	**.157**	**.258**	.204	.288	.184	.289	.212	.299	.390	.456	.262	.364	.198	.285
ECL$_{P192}$	.181	.282	**.173**	**.272**	.208	.294	.204	.307	.213	.303	.403	.463	.34	.421	.200	.288
ECL$_{P336}$	.197	.296	**.185**	**.284**	.224	.310	.217	.320	.228	.317	.417	.466	.624	.608	.214	.302
ECL$_{P720}$	.231	.324	**.219**	**.314**	.265	.341	.284	.363	.270	.348	.439	.483	.758	.687	.254	.333
ETTH1$_{P96}$	.386	.409	.390	.411	**.382**	**.399**	.478	.448	.389	.400	.867	.702	.505	.479	.396	.413
ETTH1$_{P192}$	**.429**	.436	.432	.438	.431	**.426**	.561	.504	.440	.43	.931	.751	.823	.601	.458	.448
ETTH1$_{P336}$	**.466**	.457	.480	.460	.476	**.449**	.612	.537	.482	.453	.96	.763	.731	.580	.508	.472
ETTH1$_{P720}$	.494	.483	.542	.508	.495	.487	.601	.541	**.486**	**.479**	.994	.782	.699	.590	.546	.503
EXC.$_{P192}$	.243	.351	.200	.320	**.175**	**.297**	.259	.370	.178	.301	1.22	.916	.306	.409	.177	.300
EXC.$_{P336}$	.431	.476	.346	.425	**.322**	**.409**	.478	.501	.328	.415	1.22	.917	.462	.508	.326	.414
ILI$_{P60}$	**1.99**	**.878**	2.372	.945	1.99	.905	2.367	.966	2.307	.970	4.791	1.46	3.812	1.33	1.90	.868
TRAF.$_{P96}$	.47	.318	**.465**	**.298**	.606	.389	.611	.336	.643	.405	.845	.465	.744	.452	.524	.351
TRAF.$_{P192}$	.485	.323	**.484**	**.306**	.592	.382	.643	.352	.603	.387	.883	.477	1.09	.638	.519	.346
TRAF.$_{P336}$	.497	.325	**.494**	**.312**	.600	.384	.662	.363	.612	.389	.907	.488	1.19	.692	.530	.350
TRAF.$_{P720}$	**.53**	.34	.534	**.335**	.633	.401	.678	.365	.652	.406	.974	.522	1.34	.761	.562	.366
WEA.$_{P96}$	.158	.208	**.157**	**.206**	.193	.232	.169	.220	.194	.233	.239	.323	.251	.315	.182	.222
WEA.$_{P192}$	**.207**	.253	.208	**.251**	.238	.269	.223	.264	.238	.268	.323	.399	.289	.335	.228	.261
WEA.$_{P336}$	**.264**	.294	**.264**	**.291**	.291	.306	.279	.302	.290	.304	.333	.386	.329	.356	.282	.299
WEA.$_{P720}$	**.341**	**.344**	.344	**.344**	.365	.354	.359	.355	.363	.35	.424	.447	.39	.387	.359	.349
BEST COUNT	8/20	2/20	**9/20**	**12/20**	3/20	5/20	0/20	1/20	1/20	1/20	0/20	0/20	0/20	0/20	1/20	1/20
AVERAGE	**.439**	.381	.453	**.376**	.466	.394	.525	.412	.488	.401	.931	.623	.809	.571	.449	.386
SHARED	✓	✓	✓	✓	×	×	×	×	×	×	×	×	×	×	×	×

MULTI-TASK CLASSIFICATION (ACCURACY↑)

CLASS. /NUM.	UNITS -*SUP*	UNITS -*PMT*	ITRA. [66]	TIM. [80]	PAT. [80]	PYRA. [64]	AUT. [111]	GPT. [126]
2/7	**73.1**	**73.1**	72.4	73.0	70.8	61.5	66.2	73.1
3/1	79.7	**81.4**	79.4	78.0	79.2	**81.4**	69.9	79.4
4/1	96.0	**99.0**	79.0	91.0	77.0	74.0	60.0	96.0
5/1	92.8	92.4	93.3	92.6	**94.3**	91.4	91.9	93.0
6/1	95.1	**95.8**	93.6	90.6	75.8	88.7	30.2	96.2
7/2	72.7	72.6	70.2	63.5	71.6	**74.3**	67.7	71.1
8/1	82.2	**85.3**	82.2	84.4	81.9	72.2	42.2	81.9
9/1	92.2	90.3	95.9	**97.6**	94.1	85.4	94.1	94.6
10/2	92.2	89.7	93.5	**97.2**	88.9	72.2	86.1	95.8
52/1	**89.6**	80.8	88.2	88.9	86.5	21.4	21.7	89.7
BEST	3/18	**7/18**	0/18	4/18	3/18	4/18	0/18	2/18
AVG.	**81.6**	81.2	80.3	80.9	78.1	68.8	65.6	82.0
SHARED	✓	✓	×	×	×	×	×	×

2. UNITS

2-4. UniTS Model

(5) Experiments

d) Forecasting new horizons

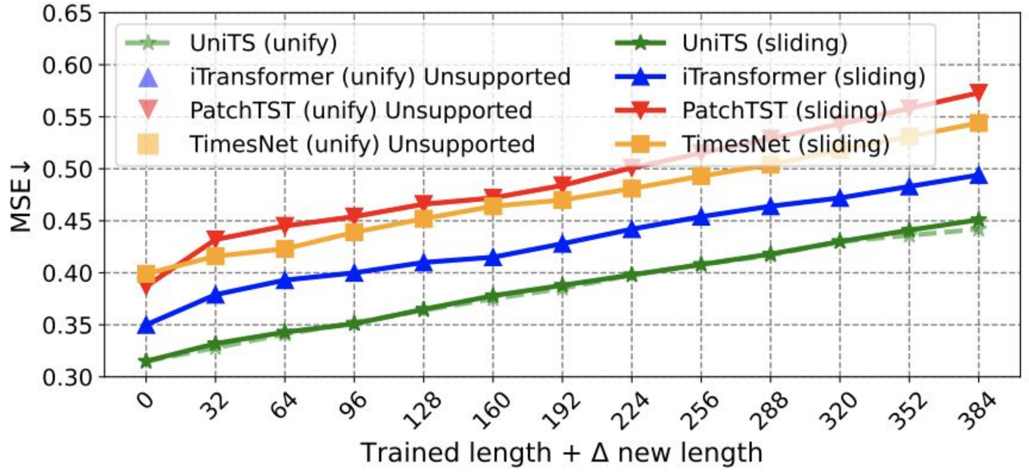


Figure 3: Zero-shot forecasting to new forecasting lengths. UNITS achieves any new forecasting length with unified direct multi-step inference. Baseline methods use the sliding windows inference as they do not support direct multi-step inference.

New forecasting horizons for baselines?

- **Sliding-window approach**

2. UNITS

2-4. UniTS Model

(5) Experiments

d) Few-shot

d-1) Forecasting

d-2) Classification

d-3) Imputation

d-4) Anomaly detection

Table 3: Few-shot multi-task learning on 9 forecasting and 6 classification tasks on out-of-domain datasets. Ratio is the data ratio of the dataset used for training. Full results in Table 24.

Model	Ratio	Acc↑	MSE↓	MAE↓	Best Count	Shared
iTransformer-*FT*	5%	56.4	0.598	0.487	1/24	×
UNITS-*PMT*	5%	55.7	**0.508**	**0.440**	16/24	✓
UNITS-*FT*	5%	**57.4**	0.530	0.448	7/24	✓
iTransformer-*FT*	15%	56.5	0.524	0.447	4/24	×
UNITS-*PMT*	15%	59.5	0.496	0.435	4/24	✓
UNITS-*FT*	15%	**61.8**	**0.487**	**0.428**	16/24	✓
iTransformer-*FT*	20%	59.9	0.510	0.438	4/24	×
UNITS-*PMT*	20%	63.6	0.494	0.435	3/24	✓
UNITS-*FT*	20%	**65.2**	**0.481**	**0.425**	17/24	✓

OOD forecasting & classification datasets

Table 8: Datasets for few-shot learning on classification and forecasting tasks. Prediction length or number of classes are indicated in parenthesis for Forecast and Classification respectively.

Name	Train Size	Sequence Length	Variables	Task	Class
ECG200 [82]	100	96	1	Classification (2)	ECG
SelfRegulationSCP1 [7]	268	896	6	Classification (2)	EEG
RacketSports [4]	151	30	6	Classification (4)	Human Activity
Handwriting [92]	150	152	3	Classification (26)	Human Activity
Epilepsy [102]	137	207	3	Classification (4)	Human Activity
StarLightCurves [89]	1000	1024	1	Classification (3)	Sensor
ETTh2$_{P96}$ [124]	8449	96	7	Forecast (96)	Electricity
ETTh2$_{P192}$ [124]	8353	96	7	Forecast (192)	Electricity
ETTh2$_{P336}$ [124]	8209	96	7	Forecast (336)	Electricity
ETTh2$_{P720}$ [124]	7825	96	7	Forecast (720)	Electricity
ETTm1$_{P96}$ [124]	34369	96	7	Forecast (96)	Electricity
ETTm1$_{P192}$ [124]	34273	96	7	Forecast (192)	Electricity
ETTm1$_{P336}$ [124]	34129	96	7	Forecast (336)	Electricity
ETTm1$_{P720}$ [124]	33745	96	7	Forecast (720)	Electricity
SaugeenRiverFlow [72]	18921	48	1	Forecast (24)	Weather

2. UNITS

2-4. UniTS Model

(5) Experiments

d) Few-shot

d-1) Forecasting

d-2) Classification

d-3) Imputation

d-4) Anomaly detection

Table 4: Few-shot multi-task learning for block-wise imputation on 6 datasets. Full results are in Table 23.

Impu. (MSE)	Ratio	ECL	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Avg	Best	Shared
TimesNet-*FT*	25%	0.245	0.369	0.193	0.442	0.119	0.106	0.246	0/6	×
	50%	0.258	0.412	0.211	0.607	0.140	0.125	0.292	0/6	×
PatchTST-*FT*	25%	0.195	0.315	0.147	0.309	0.092	0.089	0.191	0/6	×
	50%	0.230	0.353	0.175	0.442	0.111	0.105	0.236	0/6	×
iTrans-*FT*	25%	0.174	0.301	0.185	0.254	0.113	0.087	0.186	0/6	×
	50%	0.203	0.332	0.205	0.372	0.136	0.106	0.226	0/6	×
UNITS-*PMT*	25%	**0.117**	0.281	**0.177**	0.247	0.095	0.075	0.165	2/6	✓
	50%	**0.135**	0.323	**0.246**	0.343	0.131	**0.093**	0.212	3/6	✓
UNITS-*FT*	25%	0.143	**0.277**	0.194	**0.204**	**0.088**	**0.074**	**0.163**	**4/6**	✓
	50%	0.161	**0.313**	0.252	**0.295**	**0.119**	0.096	**0.206**	**3/6**	✓

Table 5: Few-shot multi-task learning on anomaly detection tasks on 5 datasets.

Anomaly (F1↑)	MSL	PSM	SMAP	SMD	SWAT	Avg	Best	Shared
Anomaly Trans.	78.0	90.2	68.3	77.8	81.5	79.2	0/5	×
TimesNet-*FT*	33.9	91.0	68.5	84.0	**93.4**	74.2	1/5	×
iTransfomer-*FT*	80.4	96.5	67.2	82.4	89.0	83.1	0/5	×
PatchTST-*FT*	79.9	96.6	68.7	83.8	92.6	84.3	0/5	×
UNITS-*PMT*	75.4	95.5	65.8	82.3	92.5	82.3	0/5	✓
UNITS-*FT*	**81.2**	**97.3**	**76.0**	**84.7**	92.5	**86.3**	**4/5**	✓

2. UNITS

2-5. Conclusion

UNITS

- **(1) Unified model** for TS analysis
  - Employs a **(2) universal specification of TS tasks**
- Effectively handles **(3) multi-domain TS data** with heterogeneous representations
- Experiments
  - Outperforming task-specific models and reprogrammed LLMs
    - on **(4) 38 multi-domain and multi-task** datasets
- (5) Strong **(5) zero-shot, few-shot, and prompt-based** performance

3. Future Works

3. Future Works

2-5. Conclusion

How to handle multi-domain datasets effectively?

=> Analyze in terms of ***"Channel Dependence"***

3. Future Works

2-5. Conclusion

How to handle multi-domain datasets effectively?

=> Analyze in terms of ***"Channel Dependence"***

- Channel Dependence (CD)
    - Considers the interactions btw the channels when embedding TS
- Channel Independence (CI)
    - Does NOT considers the interactions btw the channels when embedding TS

Adaptive CD/CI strategies for dataset/tasks!

Thank You!