# A Experimental Details

## A.1 Reproduction details for Ti-MAE

The default setting and all the components in details of Ti-MAE is shown in Table 1. We use one Conv1d layer with the setting of kernel $= 3$, stride $= 1$, padding $= 1$ to obtain the encoder input embedding, and then we add a fixed positional encoding as

$$\begin{aligned} \text{PE}(pos, 2i) &= \sin(\frac{pos}{10000^{2i/\text{d}_{\text{model}}}}) \\ \text{PE}(pos, 2i+1) &= \cos(\frac{pos}{10000^{2i/\text{d}_{\text{model}}}}), \end{aligned} \tag{1}$$

where $\text{d}_{\text{model}}$ represents the hidden state dimension. After encoder input embedding, we randomly mask out 75% tokens and remaining visible parts are fed into the encoder. The encoder and decoder of Ti-MAE both contain 2 Transformer blocks, each of which consists of one vanilla self-attention layer with 4 heads and a point-wise feed forward layer. Similar as ViT Dosovitskiy et al. (2021), we adopt pre-norm instead of post-norm for stability of the model. In the decoder, we first apply a linear layer to reduce the input dimension ($64 \rightarrow 32$). Given the position to be reconstruct, randomly initialized learnable masked tokens are padded to the encoded tokens with the original positional encoding. A dropout layer ($p = 0.1$) is added to the bottom of Transformer blocks to prevent the over-fitting problem. The last linear projection layer of the decoder is to reconstruct the missing values at the point-level. All the linear layers in Ti-MAE are initialized through xavier Glorot & Bengio (2010). Note that the choice of Transformer blocks in Ti-MAE is flexible so that you can use other designed Transformer blocks with more inductive biases if necessary.

Table 1: Default setting of Ti-MAE

| Config | Value |
|---|---|
| optimizer | Adam Kingma & Ba (2015) |
| learning rate | 0.001 |
| learning rate schedule | cosine decay |
| epochs | 10 |
| masking ratio | 75% |
| sampling time | 30 |
| batch size | 64 |
| #encoder layer | 2 |
| #decoder layer | 2 |
| $\text{d}_{\text{model}}$ | 64 |
| dropout | 0.1 |

## A.2 Details on baselines

For forecasting tasks, the results of CoST Woo et al. (2022), TS2Vec Yue et al. (2022), TNC Tonekaboni et al. (2021), MoCo Chen et al. (2021), Autoformer Wu et al. (2021), Informer Zhou et al. (2021), LogTrans LI et al. (2019) and TCN Bai et al. (2018) are all based on our reproduction. For classification tasks, the results of TS2Vec are based on our reproduction. Other results of classification are directly taken from Yue et al. (2022).

CoST Woo et al. (2022) was recently proposed as a contrastive learning framework of disentangled seasonal-trend representations for time series forecasting. They comprises both time domain and frequency domain contrastive losses to learn discriminative trend and seasonal representations. We use the public official source code from `https://github.com/salesforce/CoST`.

TS2Vec Yue et al. (2022) is a universal framework for learning representations of time series in an arbitrary semantic level through applying contrastive learning in a hierarchical way over augmented context views. TS2Vec can obtain timestamp-level and instance-level representations for forecasting and classification simultaneously. We take the officially implemented code from `https://github.com/yuezhihan/ts2vec`.

TNC Tonekaboni et al. (2021) is a self-supervised contrastive learning framework for time series, where the positive samples come from the neighboring similar signals. We use the official open source code from `https://github.com/sanatonek/TNCrepresentationlearning` and all the settings of hyper-parameters follows Woo et al. (2022).

MoCo Chen et al. (2021) is a self-supervised contrastive learning framework widely used in computer vision domain, which uses a dynamic queue to save a large number of positive and negative samples with consistency. We directly apply this framework on time series data using the official code from `https://github.com/facebookresearch/moco`. Hyper-parameters are the same as Woo et al. (2022).

Autoformer Wu et al. (2021) is a novel end-to-end supervised model with a decomposition architecture for time series forecasting. By directly subtracting trend parts obtained from moving average, they design an auto-correlation mechanism as a replacement for self-attention to capture long-term dependencies from seasonality parts. We use their open source code from `https://github.com/thuml/Autoformer`. Hyper-parameters are remain the default values in the code.

Informer Zhou et al. (2021) is an efficient end-to-end supervised model for time series forecasting. They propose a novel sparse attention to reduce time complexity and memory usage. We take the officially implemented code from `https://github.com/zhouhaoyi/Informer2020`. Hyper-parameters are used as suggested in their paper.

LogTrans LI et al. (2019) proposes a novel sparse attention scheme compared to canonical Transformer and reduces the usage of computing resources. Due to no official code available, we follow Yue et al. (2022) and refer to their experimental settings.

TCN Bai et al. (2018) is a strong feature extractor based on dilated convolutions for time series, and is also widely used as backbone in other frameworks. We use the official code from `https://github.com/locuslab/TCN` and follow Yue et al. (2022) to set all the hyper-parameters.

### A.3 Details on benchmark tasks

For time series forecasting tasks, the evaluation settings of end-to-end supervised models and other representation learning methods are slightly different. For other representation learning methods, we follow Yue et al. (2022) to evaluate the performance of their models. Specifically, we use a ridge regression trained on the learned representations to predict the future values. The regularization term $\alpha$ is selected by grid search from $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$. For end-to-end models, we set the sum of the input length and the predicted length as a constant value (*e.g.* 200 length input to predict 100 length future values) for fairness. We set the size of the observation window to 300 to evaluate forecasting performance at different time steps.

For classification tasks, we directly obtain instance-level representations by average or max pooling over all timestamps following Yue et al. (2022). To evaluate the performance of models on classification, we follow the same protocol Franceschi et al. (2019), where an SVM classifier with RBF kernel is trained on obtained instance-level representations. The full results of each dataset in UCR are provided in Table 6 and 7.

Notably, due to the flexible design of the Transformer block, we can utilize any layer of the encoder or the decoder of Ti-MAE to get intermediate representations. Extra class token is also a choice if necessary. In our experiments, we simply gather the encoder embedding of Ti-MAE as instance-level representations for evaluation. To accelerate the training of the model, we perform equidistant sampling for different datasets to reduce input length.

TS2Vec also reports an interesting phenomenon that using Transformer instead of Dilated CNN as backbone will largely degrade the performance on classification tasks. We also find similar problems, especially on morphological datasets. We suppose that some morphological datasets have almost no seasonality, while the local morphological characteristics determine the data classification. The positional encoding introduced in the encoder may destroy these morphological features. Simply removing position embedding in the encoder when generating representations will significantly affect the performance of classification. Table 2 shows the classification results on some morphological datasets with or without position embedding.

Table 2: The classification results on morphological datasets with or without positional encoding

| dataset | w/ PE | w/o PE |
|---|---|---|
| Beef | **0.90** | 0.77 |
| OSULeaf | 0.59 | **0.74** |
| ShapeletSim | 0.54 | **0.91** |
| Worms | 0.64 | **0.78** |

# B Additional Experimental Results

## B.1 The impact of the length of historical time series

Table 3 shows the time series forecasting results on Weather with different length of input historical data. Longer input time series will significantly degrade the performance of other Transformer-based end-to-end models, which is in contrast to the ability of Transformer to capture long-term dependencies. We suppose that it is the fixed training paradigm that limits the ability of Transformer-based models to learn adequate local and global features.

Table 3: The results of forecasting 100 time steps in future on Weather with different length of input.

| Input length | | 100 | 200 | 400 |
|---|---|---|---|---|
| Ti-MAE | MSE | 0.2413 | **0.2103** | **0.2328** |
| | MAE | **0.2844** | **0.2696** | **0.2985** |
| Autoformer | MSE | **0.2375** | 0.3143 | 0.3144 |
| | MAE | 0.3127 | 0.3789 | 0.3810 |

## B.2 Ablation study on Ti-MAE's components

Table 4 shows the impact of different components of Ti-MAE, which proves the effectiveness of random masking strategy, Transformer-based backbone and other necessary parts.

Table 4: Ablation study of Ti-MAE's components on the Exchange dataset.

| Ablation variant | MSE | MAE |
|---|---|---|
| Default | **0.2111** | **0.3367** |
| Random $\rightarrow$ fixed continuous masking | 0.2505 (-18.7%) | 0.3700 (-9.9%) |
| Encoder w/o positional encoding | 0.3082 (-46.0%) | 0.3996 (-18.7%) |
| Decoder w/o positional encoding | 0.2276 (-7.8%) | 0.3518 (-4.5%) |
| pre-norm $\rightarrow$ post-norm | 0.2213 (-4.8%) | 0.3474 (-3.2%) |
| Transformer $\rightarrow$ TCN | 0.2340 (-10.8%) | 0.3558 (-5.7%) |
| Transformer $\rightarrow$ LSTM | 0.2406 (-14.0%) | 0.3612 (-7.3%) |

## B.3 Transferability study

To evaluate the transferability of our framework, we generate a set of time series data with different trend and seasonality patterns, which follows

$$y(t) = \cos(\alpha \cdot t) + \cos(\frac{\alpha}{2} \cdot t) + \cos(\frac{\alpha}{4} \cdot t) + \beta \cdot t + \epsilon \tag{2}$$

where the hyper-parameters $\alpha$ and $\beta$ respectively control the trend and seasonality patterns, and the noises $\epsilon \sim N(0, 0.1)$. We train our Ti-MAE under the setting of $\alpha = 300, \beta = 3$ and evaluate the forecasting performance on other different settings. Table 5 and Figure 1 demonstrate the strong transferability of Ti-MAE under different trend and seasonality patterns.

Table 5: The results of forecasting 400 time steps on simulated time series data with different trend and seasonality patterns.

| Setting | $\alpha = 300$ $\beta = 3$ | $\alpha = 600$ $\beta = 3$ | $\alpha = 300$ $\beta = 100$ | $\alpha = 600$ $\beta = 100$ |
|---|---|---|---|---|
| MSE | 0.0134 | 0.0596 | 0.0089 | 0.0232 |
| MAE | 0.0778 | 0.1912 | 0.0881 | 0.0711 |

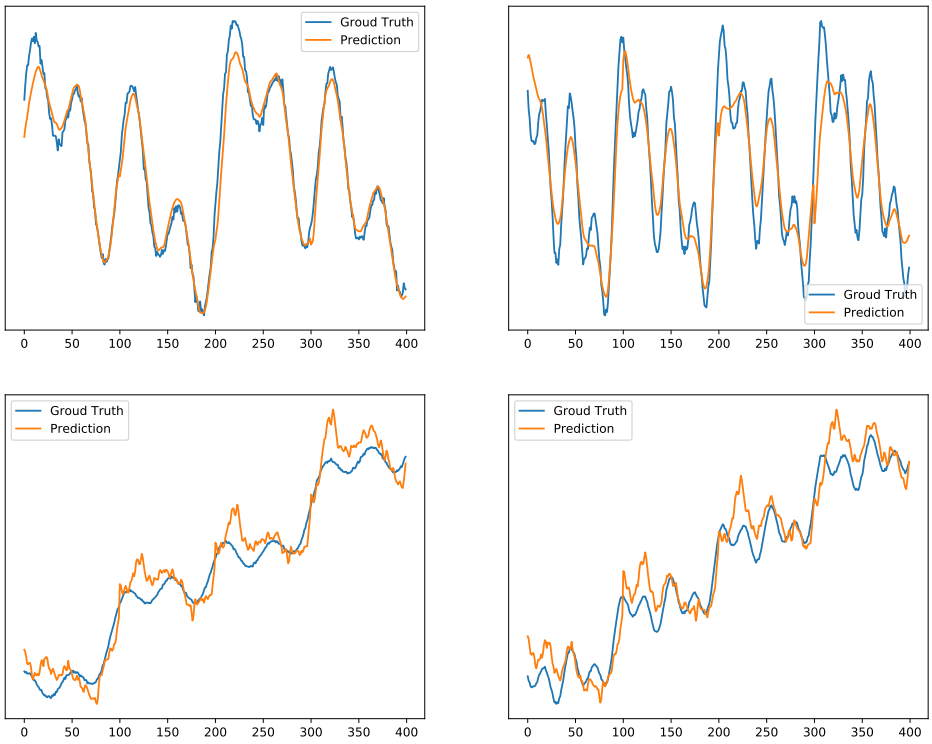


Figure 1: Transferability of Ti-MAE on different trend and seasonality patterns.

Table 6: Full classification results on 128 UCR datasets part 1.

| Dataset | TMAE | TS2Vec | T-Loss | TNC | TS-TCC | TST | DTW |
|---|---|---|---|---|---|---|---|
| ACSF1 | 0.820 | 0.870 | **0.900** | 0.730 | 0.730 | 0.760 | 0.640 |
| Adiac | **0.788** | 0.726 | 0.675 | 0.726 | 0.767 | 0.550 | 0.604 |
| AllGestureWiimoteX | 0.633 | **0.782** | 0.763 | 0.703 | 0.697 | 0.259 | 0.716 |
| AllGestureWiimoteY | 0.682 | **0.791** | 0.726 | 0.699 | 0.741 | 0.423 | 0.729 |
| AllGestureWiimoteZ | 0.671 | **0.760** | 0.723 | 0.646 | 0.689 | 0.447 | 0.643 |
| ArrowHead | **0.874** | 0.794 | 0.766 | 0.703 | 0.737 | 0.771 | 0.703 |
| BME | **1.000** | 0.987 | 0.993 | 0.973 | 0.933 | 0.760 | 0.900 |
| Beef | **0.900** | 0.700 | 0.667 | 0.733 | 0.600 | 0.500 | 0.633 |
| BeetleFly | 0.900 | 0.750 | 0.800 | 0.850 | 0.800 | **1.000** | 0.700 |
| BirdChicken | **1.000** | 0.800 | 0.850 | 0.750 | 0.650 | 0.650 | 0.750 |
| CBF | **1.000** | **1.000** | 0.983 | 0.983 | 0.998 | 0.898 | 0.997 |
| Car | **0.867** | 0.800 | 0.833 | 0.683 | 0.583 | 0.55 | 0.733 |
| Chinatown | **0.985** | 0.974 | 0.951 | 0.977 | 0.983 | 0.936 | 0.957 |
| ChlorineConcentration | 0.725 | **0.804** | 0.749 | 0.760 | 0.753 | 0.562 | 0.648 |
| CinCECGTorso | **0.971** | 0.793 | 0.713 | 0.669 | 0.671 | 0.508 | 0.651 |
| Coffee | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.821 | **1.000** |
| Computers | **0.780** | 0.648 | 0.664 | 0.684 | 0.704 | 0.696 | 0.700 |
| CricketX | 0.674 | **0.777** | 0.713 | 0.623 | 0.731 | 0.385 | 0.754 |
| CricketY | 0.659 | **0.769** | 0.728 | 0.597 | 0.718 | 0.467 | 0.744 |
| CricketZ | 0.718 | **0.810** | 0.708 | 0.682 | 0.713 | 0.403 | 0.754 |
| Crop | 0.751 | **0.756** | 0.722 | 0.738 | 0.742 | 0.710 | 0.665 |
| DiatomSizeReduction | 0.984 | 0.990 | 0.984 | **0.993** | 0.977 | 0.961 | 0.967 |
| DistalPhalanxOutlineAgeGroup | 0.763 | 0.719 | 0.727 | 0.741 | 0.755 | 0.741 | **0.770** |
| DistalPhalanxOutlineCorrect | **0.793** | 0.754 | 0.775 | 0.754 | 0.754 | 0.728 | 0.717 |
| DistalPhalanxTW | **0.727** | 0.698 | 0.676 | 0.669 | 0.676 | 0.568 | 0.590 |
| DodgerLoopDay | **0.613** | 0.538 | 0.241 | 0.183 | 0.206 | 0.200 | 0.500 |
| DodgerLoopGame | 0.739 | **0.826** | 0.415 | 0.508 | 0.493 | 0.696 | 0.877 |
| DodgerLoopWeekend | **0.978** | 0.949 | 0.623 | 0.684 | 0.601 | 0.732 | 0.949 |
| ECG200 | 0.910 | 0.860 | **0.940** | 0.830 | 0.880 | 0.830 | 0.770 |
| ECG5000 | **0.942** | 0.932 | 0.933 | 0.937 | 0.941 | 0.928 | 0.924 |
| ECGFiveDays | 0.988 | **1.000** | **1.000** | 0.999 | 0.878 | 0.763 | 0.768 |
| EOGHorizontalSignal | 0.558 | 0.528 | **0.605** | 0.442 | 0.401 | 0.373 | 0.503 |
| EOGVerticalSignal | **0.547** | 0.483 | 0.434 | 0.392 | 0.376 | 0.298 | 0.448 |
| Earthquakes | **0.748** | **0.748** | **0.748** | **0.748** | **0.748** | **0.748** | 0.719 |
| ElectricDevices | 0.685 | **0.724** | 0.707 | 0.700 | 0.686 | 0.676 | 0.602 |
| EthanolLevel | **0.744** | 0.388 | 0.382 | 0.424 | 0.486 | 0.260 | 0.276 |
| FaceAll | **0.880** | 0.789 | 0.786 | 0.766 | 0.813 | 0.504 | 0.808 |
| FaceFour | 0.875 | 0.852 | **0.920** | 0.659 | 0.773 | 0.511 | 0.830 |
| FacesUCR | 0.866 | **0.929** | 0.884 | 0.789 | 0.863 | 0.543 | 0.905 |
| FiftyWords | **0.787** | 0.754 | 0.732 | 0.653 | 0.653 | 0.525 | 0.690 |
| Fish | 0.897 | **0.920** | 0.891 | 0.817 | 0.817 | 0.720 | 0.823 |
| FordA | 0.818 | **0.940** | 0.928 | 0.902 | 0.930 | 0.568 | 0.555 |
| FordB | 0.652 | 0.802 | 0.793 | 0.733 | **0.815** | 0.507 | 0.620 |
| FreezerRegularTrain | 0.987 | 0.984 | 0.956 | **0.991** | 0.989 | 0.922 | 0.899 |
| FreezerSmallTrain | 0.959 | 0.872 | 0.933 | 0.982 | **0.979** | 0.920 | 0.753 |
| Fungi | 0.968 | 0.935 | **1.000** | 0.527 | 0.753 | 0.366 | 0.839 |
| GestureMidAirD1 | **0.662** | 0.592 | 0.608 | 0.431 | 0.369 | 0.208 | 0.569 |
| GestureMidAirD2 | **0.546** | 0.523 | **0.546** | 0.362 | 0.254 | 0.138 | 0.608 |
| GestureMidAirD3 | **0.400** | 0.323 | 0.285 | 0.292 | 0.177 | 0.154 | 0.323 |
| GesturePebbleZ1 | **0.901** | 0.849 | 0.919 | 0.378 | 0.395 | 0.500 | 0.791 |
| GesturePebbleZ2 | **0.918** | 0.854 | 0.899 | 0.316 | 0.430 | 0.380 | 0.671 |
| GunPoint | **0.993** | 0.973 | 0.980 | 0.967 | **0.993** | 0.827 | 0.907 |
| GunPointAgeSpan | **0.994** | 0.962 | **0.994** | 0.984 | **0.994** | 0.991 | 0.918 |
| GunPointMaleVersusFemale | 0.997 | **1.000** | 0.997 | 0.994 | 0.997 | **1.000** | 0.997 |
| GunPointOldVersusYoung | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.838 |
| Ham | **0.800** | 0.714 | 0.724 | 0.752 | 0.743 | 0.524 | 0.467 |
| HandOutlines | 0.919 | 0.919 | 0.922 | **0.930** | 0.724 | 0.735 | 0.881 |
| Haptics | 0.484 | **0.519** | 0.490 | 0.474 | 0.396 | 0.357 | 0.377 |
| Herring | **0.656** | 0.609 | 0.594 | 0.594 | 0.594 | 0.594 | 0.531 |
| HouseTwenty | **0.941** | 0.899 | 0.933 | 0.782 | 0.790 | 0.815 | 0.924 |
| InlineSkate | 0.380 | **0.403** | 0.371 | 0.378 | 0.347 | 0.287 | 0.384 |
| InsectEPGRegularTrain | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.872 |
| InsectEPGSmallTrain | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | **1.000** | 0.735 |
| InsectWingbeatSound | **0.639** | 0.616 | 0.597 | 0.549 | 0.415 | 0.266 | 0.355 |

Table 7: Full classification results on 128 UCR datasets part 2.

| Dataset | TMAE | TS2Vec | T-Loss | TNC | TS-TCC | TST | DTW |
|---|---|---|---|---|---|---|---|
| ItalyPowerDemand | **0.967** | 0.932 | 0.954 | 0.928 | 0.955 | 0.845 | 0.950 |
| LargeKitchenAppliances | 0.787 | **0.869** | 0.789 | 0.776 | 0.848 | 0.595 | 0.795 |
| Lightning2 | 0.836 | **0.869** | **0.869** | **0.869** | 0.836 | 0.705 | **0.869** |
| Lightning7 | **0.808** | 0.781 | 0.795 | 0.767 | 0.685 | 0.411 | 0.726 |
| Mallat | **0.956** | 0.904 | 0.951 | 0.871 | 0.922 | 0.713 | 0.934 |
| Meat | **0.967** | **0.967** | 0.950 | 0.917 | 0.883 | 0.900 | 0.933 |
| MedicalImages | 0.771 | **0.799** | 0.750 | 0.754 | 0.747 | 0.632 | 0.737 |
| MelbournePedestrian | 0.949 | **0.958** | 0.944 | 0.942 | 0.949 | 0.741 | 0.791 |
| MiddlePhalanxOutlineAgeGroup | **0.675** | 0.643 | 0.656 | 0.643 | 0.630 | 0.617 | 0.500 |
| MiddlePhalanxOutlineCorrect | 0.811 | **0.831** | 0.825 | 0.818 | 0.818 | 0.753 | 0.698 |
| MiddlePhalanxTW | **0.623** | 0.578 | 0.591 | 0.571 | 0.610 | 0.506 | 0.506 |
| MixedShapesRegularTrain | **0.922** | 0.917 | 0.905 | 0.911 | 0.855 | 0.879 | 0.842 |
| MixedShapesSmallTrain | **0.875** | 0.854 | 0.860 | 0.813 | 0.735 | 0.828 | 0.780 |
| MoteStrain | **0.913** | 0.859 | 0.851 | 0.825 | 0.843 | 0.768 | 0.835 |
| NonInvasiveFetalECGThorax1 | 0.918 | **0.924** | 0.878 | 0.898 | 0.898 | 0.471 | 0.790 |
| NonInvasiveFetalECGThorax2 | 0.938 | **0.939** | 0.919 | 0.912 | 0.913 | 0.832 | 0.865 |
| OSULeaf | 0.736 | **0.851** | 0.760 | 0.723 | 0.723 | 0.545 | 0.591 |
| OliveOil | **0.933** | 0.867 | 0.867 | 0.833 | 0.800 | 0.800 | 0.833 |
| PLAID | 0.458 | 0.555 | 0.555 | 0.495 | 0.445 | 0.419 | **0.840** |
| PhalangesOutlinesCorrect | 0.772 | **0.806** | 0.784 | 0.787 | 0.804 | 0.773 | 0.728 |
| Phoneme | 0.229 | **0.296** | 0.276 | 0.180 | 0.242 | 0.139 | 0.228 |
| PickupGestureWiimoteZ | **0.840** | 0.800 | 0.740 | 0.620 | 0.600 | 0.240 | 0.660 |
| PigAirwayPressure | 0.240 | **0.807** | 0.510 | 0.413 | 0.380 | 0.120 | 0.106 |
| PigArtPressure | 0.760 | **0.966** | 0.928 | 0.808 | 0.524 | 0.774 | 0.245 |
| PigCVP | 0.750 | **0.813** | 0.788 | 0.649 | 0.615 | 0.596 | 0.154 |
| Plane | **1.000** | 0.990 | 0.990 | **1.000** | **1.000** | 0.933 | **1.000** |
| PowerCons | **1.000** | 0.967 | 0.900 | 0.933 | 0.961 | 0.911 | 0.878 |
| ProximalPhalanxOutlineAgeGroup | **0.863** | 0.834 | 0.844 | 0.854 | 0.839 | 0.854 | 0.805 |
| ProximalPhalanxOutlineCorrect | 0.876 | **0.890** | 0.859 | 0.866 | 0.873 | 0.770 | 0.784 |
| ProximalPhalanxTW | **0.829** | 0.790 | 0.771 | 0.810 | 0.800 | 0.780 | 0.761 |
| RefrigerationDevices | **0.611** | 0.603 | 0.515 | 0.565 | 0.563 | 0.483 | 0.464 |
| Rock | 0.660 | 0.660 | 0.580 | 0.580 | 0.600 | **0.680** | 0.600 |
| ScreenType | **0.579** | 0.411 | 0.416 | 0.509 | 0.419 | 0.419 | 0.397 |
| SemgHandGenderCh2 | 0.838 | **0.960** | 0.890 | 0.882 | 0.837 | 0.725 | 0.802 |
| SemgHandMovementCh2 | 0.700 | **0.862** | 0.789 | 0.593 | 0.613 | 0.420 | 0.584 |
| SemgHandSubjectCh2 | 0.813 | **0.947** | 0.853 | 0.771 | 0.753 | 0.484 | 0.727 |
| ShakeGestureWiimoteZ | 0.900 | **0.940** | 0.920 | 0.820 | 0.860 | 0.760 | 0.860 |
| ShapeletSim | 0.911 | **0.939** | 0.672 | 0.589 | 0.683 | 0.489 | 0.650 |
| ShapesAll | 0.840 | **0.890** | 0.848 | 0.788 | 0.773 | 0.733 | 0.768 |
| SmallKitchenAppliances | **0.741** | 0.733 | 0.677 | 0.725 | 0.691 | 0.592 | 0.643 |
| SmoothSubspace | **0.993** | 0.980 | 0.960 | 0.913 | 0.953 | 0.827 | 0.827 |
| SonyAIBORobotSurface1 | **0.912** | 0.910 | 0.902 | 0.804 | 0.899 | 0.724 | 0.725 |
| SonyAIBORobotSurface2 | **0.934** | 0.897 | 0.889 | 0.834 | 0.907 | 0.745 | 0.831 |
| StarLightCurves | **0.972** | 0.971 | 0.964 | 0.968 | 0.967 | 0.949 | 0.907 |
| Strawberry | **0.970** | 0.967 | 0.954 | 0.951 | 0.965 | 0.916 | 0.941 |
| SwedishLeaf | **0.938** | 0.923 | 0.914 | 0.880 | 0.923 | 0.738 | 0.792 |
| Symbols | 0.961 | **0.981** | 0.963 | 0.885 | 0.916 | 0.786 | 0.950 |
| SyntheticControl | 0.993 | 0.997 | 0.987 | **1.000** | 0.990 | 0.490 | 0.993 |
| ToeSegmentation1 | 0.890 | 0.925 | **0.939** | 0.864 | 0.930 | 0.807 | 0.772 |
| ToeSegmentation2 | **0.908** | 0.900 | 0.900 | 0.831 | 0.877 | 0.615 | 0.838 |
| Trace | **1.000** | **1.000** | 0.990 | **1.000** | **1.000** | **1.000** | **1.000** |
| TwoLeadECG | 0.985 | 0.982 | **0.999** | 0.993 | 0.976 | 0.871 | 0.905 |
| TwoPatterns | 0.994 | **1.000** | 0.999 | **1.000** | 0.999 | 0.466 | **1.000** |
| UMD | **1.000** | 0.993 | 0.993 | 0.993 | 0.986 | 0.910 | 0.993 |
| UWaveGestureLibraryAll | **0.956** | 0.938 | 0.896 | 0.903 | 0.692 | 0.475 | 0.892 |
| UWaveGestureLibraryX | **0.814** | 0.797 | 0.785 | 0.781 | 0.733 | 0.569 | 0.728 |
| UWaveGestureLibraryY | **0.736** | 0.714 | 0.710 | 0.697 | 0.641 | 0.348 | 0.634 |
| UWaveGestureLibraryZ | 0.749 | **0.759** | 0.757 | 0.721 | 0.690 | 0.655 | 0.658 |
| Wafer | 0.996 | **0.999** | 0.992 | 0.994 | 0.994 | 0.991 | 0.980 |
| Wine | **0.907** | 0.741 | 0.815 | 0.759 | 0.778 | 0.500 | 0.574 |
| WordSynonyms | **0.705** | 0.676 | 0.691 | 0.630 | 0.531 | 0.422 | 0.649 |
| Worms | **0.779** | 0.727 | 0.727 | 0.623 | 0.753 | 0.455 | 0.584 |
| WormsTwoClass | **0.792** | 0.740 | **0.792** | 0.727 | 0.753 | 0.584 | 0.623 |
| Yoga | 0.834 | **0.888** | 0.837 | 0.812 | 0.791 | 0.830 | 0.837 |

## REFERENCES


Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS*, 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

SHIYANG LI, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *ArXiv*, abs/1907.00235, 2019.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *ArXiv*, abs/2106.00750, 2021.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PilZY3omXV2.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yu Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *AAAI*, 2022.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wan Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.