Appendix

A Proof for Observation 1

Proof. It is proved via construction. Given a simple tabular data point $\{0, 1\}$, we construct the probability distribution $P_{\mathbf{K},\mathbf{X},\mathbf{Y}}$ as follows.

$$\begin{array}{l}\Pr(\mathbf{Y}=0,\mathbf{K}=0)\\ =(1-\alpha_0)q_0, \Pr(\mathbf{Y}=0,\mathbf{K}=1)=(1-\alpha_1)q_1,\\ \Pr(\mathbf{Y}=1,\mathbf{K}=0)=\frac{\alpha_0}{2}q_0, \Pr(\mathbf{Y}=1,\mathbf{K}=1)=\frac{\alpha_1}{2}q_1,\\ \Pr(\mathbf{Y}=2,\mathbf{K}=0)=\frac{\alpha_0}{2}q_0, \Pr(\mathbf{Y}=2,\mathbf{K}=1)=\frac{\alpha_1}{2}q_1,\\ \Pr(\mathbf{Y}=1,\mathbf{K}=0)=\alpha_0 q_0, \Pr(\mathbf{Y}=0,\mathbf{K}=0)=(1-\alpha_0)\,q_0,\\ \Pr(\mathbf{Y}=1,\mathbf{K}=1)=\alpha_1 q_1, \Pr(\mathbf{Y}=0,\mathbf{K}=1)=(1-\alpha_1)\,q_1.\end{array}$$

Thus, $I(\mathbf{M};\mathbf{Y})=(1-\alpha)\log\frac{1}{1-\alpha}+\alpha\log\frac{1}{\alpha}=g(\alpha)$.

$$I(\mathbf{M};\mathbf{Y})=(1-\alpha)\log\frac{1}{1-\alpha}+\alpha\log\frac{1}{\alpha}=g(\alpha).$$

Consider a classifier $\mathcal{H}$ such that $\mathcal{H}(0)=\mathcal{H}(1)=0$, $\mathcal{H}(\text{missing})=1$, $\mathbb{E}[\mathbb{I}(\mathcal{H}(\mathbf{X})=\mathbf{Y})]=1$, and $\text{Disc}(\mathcal{H})=0$. For any ϵ, $F_\epsilon\left(P_{\mathbf{K},\mathbf{X},\mathbf{Y}}\right)=1$. Thus, we can find

$$\begin{array}{l}\Pr(\mathbf{Y}=0,\hat{\mathbf{X}}=0|\mathbf{K}=k)=\Pr(\mathbf{Y}=0,\hat{\mathbf{X}}=1|\mathbf{K}=k)=\frac{1-\alpha_k}{2},\\ \Pr(\mathbf{Y}=1,\hat{\mathbf{X}}=0|\mathbf{K}=k)=0, \Pr(\mathbf{Y}=1,\hat{\mathbf{X}}=1|\mathbf{K}=k)=\frac{\alpha_k}{2},\\ \Pr(\mathbf{Y}=2,\hat{\mathbf{X}}=0|\mathbf{K}=k)=0, \Pr(\mathbf{Y}=2,\hat{\mathbf{X}}=1|\mathbf{K}=k)=\frac{\alpha_k}{2},\\ \Pr(\mathbf{Y}=0,\hat{\mathbf{X}}=0\mid\mathbf{K}=k)\\ =\Pr(\mathbf{Y}=0,\hat{\mathbf{X}}=1\mid\mathbf{K}=k)=\frac{1-\alpha_k}{2},\\ \Pr(\mathbf{Y}=1,\hat{\mathbf{X}}=0\mid\mathbf{K}=k)=0,\\ \Pr(\mathbf{Y}=1,\hat{\mathbf{X}}=1\mid\mathbf{K}=k)=\alpha_k.\end{array}$$

We represent a probabilistic classifier $\mathcal{H}$ as

$$\begin{array}{l}\Pr(\hat{\mathbf{Y}}=0|\hat{\mathbf{X}}=0)=p_0, \Pr(\hat{\mathbf{Y}}=1|\hat{\mathbf{X}}=0)=p_1,\\ \Pr(\hat{\mathbf{Y}}=0|\hat{\mathbf{X}}=1)=p_2, \Pr(\hat{\mathbf{Y}}=1|\hat{\mathbf{X}}=1)=p_3,\\ \Pr(\hat{\mathbf{Y}}=2|\hat{\mathbf{X}}=0)=1-p_0-p_1,\\ \Pr(\hat{\mathbf{Y}}=2|\hat{\mathbf{X}}=1)=1-p_2-p_3,\end{array}$$

This ensures that $\text{Disc}(\mathcal{H})=0$. Moreover, we can obtain

$$\begin{aligned}&\mathbb{E}[\mathbb{I}(\hat{\mathbf{Y}}=\mathbf{Y})]=\Pr(\hat{\mathbf{Y}}=\mathbf{Y})\\ =&\Pr(\hat{\mathbf{Y}}=1\mid\mathbf{Y}=1,\mathbf{K}=0)\Pr(\mathbf{Y}=1,\mathbf{K}=0)\\ &+\Pr(\hat{\mathbf{Y}}=1\mid\mathbf{Y}=1,\mathbf{K}=1)\Pr(\mathbf{Y}=1,\mathbf{K}=1)\\ &+\Pr(\hat{\mathbf{Y}}=0\mid\mathbf{Y}=0,\mathbf{K}=0)\Pr(\mathbf{Y}=0,\mathbf{K}=0)\\ &+\Pr(\hat{\mathbf{Y}}=0\mid\mathbf{Y}=0,\mathbf{K}=1)\Pr(\mathbf{Y}=0,\mathbf{K}=1)\\ &=(1-p_1)(\alpha_0q_0+\alpha_1q_1)+\frac{p_0+p_1}{2}(1-\alpha_0q_0-\alpha_1q_1).\end{aligned}$$

Note that $\alpha_0q_0+\alpha_1q_1=\alpha$. Thus,

$$\begin{aligned}\max_h\mathbb{E}[\mathbb{I}(\hat{\mathbf{Y}}=\mathbf{Y})]&=\max_{p_0,p_1\in[0,1]}(1-p_1)\,\alpha+\frac{p_0+p_1}{2}(1-\alpha)\\ &=\max_{p_0,p_2\in[0,1]}\frac{(1-\alpha)}{2}p_0+\frac{(2\alpha-1)}{2}p_2+\frac{1}{2}\alpha.\end{aligned}$$

Given that $\alpha\in\left(\frac{1}{2},1\right)$, it follows that $1-\alpha>0$ and $2\alpha-1>0$. Consequently, the objective function attains its maximum value when $p_0=p_2=1$, resulting in α. In other words, $F_\epsilon(P_{\mathbf{K},\hat{\mathbf{X}},\mathbf{Y}})=\alpha$. Since $F_\epsilon\left(P_{\mathbf{K},\mathbf{X},\mathbf{Y}}\right)=1$, the corresponding equation holds. Therefore, the imputation-then-classify solution is less effective, compared to the analysis method using incomplete data representation learning (without imputation). □

Algorithm 1: The procedure of INTER

Input: the incomplete multivariate time series dataset $\mathcal{D}$ and the maximum number of model training epochs c_{max}
Output: the optimized analysis model $\mathcal{M}^*$

divide $\mathcal{D}$ into a training set $\mathcal{D}_T$ and a validation set $\mathcal{D}_V$
for *epoch* $=1$ to c_{max} **do**
  for $\mathbf{X}\in\mathcal{D}_T$ **do**
    /* **MPL module updating** */
    extract M from X
    $\mathbf{M_s}\leftarrow\text{FFN}(\mathbf{M})$
    $\mathbf{X^P}\leftarrow$ Embedding(Patching(InstanceNorm($\mathbf{X}$)))
    deploy *missing-state-aware dropping strategy* to $\mathbf{X^P}$
    $\mathbf{Z}\leftarrow MAT(concat(\mathbf{X^P},\mathbf{M_s}))$;
    /* **ITSA module updating** */
    $\mathbf{H}\leftarrow PLM(\mathbf{Z})$
    $\hat{\mathbf{X}}^T\leftarrow FL(LH(\mathbf{H}))$
    calculate the loss $\mathcal{L}_{\text{MSE}}$ or $\mathcal{L}_{\text{cls}}$ over $\mathbf{X}_T$ and $\hat{\mathbf{X}}_T$
    update $\theta_{\mathcal{M}}$ in $\mathcal{M}$ with $\mathcal{L}_{\text{MSE}}$ or $\mathcal{L}_{\text{cls}}$, $\theta_{\mathcal{M}^*}\leftarrow\theta_{\mathcal{M}}$
  if it satisfies termination criterion over $\mathcal{D}_V$ **then**
    break
return the optimized analysis model $\mathcal{M}^*$

B Pseudo-code of INTER

Our proposed INTER framework consists of two key modules: *Missing-aware Patch Learning* (MPL) and *Incomplete Time Series Analysis* (ITSA). Algorithm 1 describes the procedure of INTER.

C Proof for Theorem 1

Proof. We first introduce three different strategies: *Dropout*, *DropPatch*, and our *missing-state-aware patch dropping* (MPD), which randomly apply masking to the input time series based on certain rules, following a Bernoulli distribution. (1) *Dropout*: $\hat{m}_{i,j}=\epsilon m_{i,j}$, where $\hat{m}_{i,j}$ represents the element in the i-th row and j-th column of the perturbed mask matrix, and ϵ represents an independent and identically distributed binary random variable following a Bernoulli distribution (i.e., taking values of 0 or 1). (2) *DropPatch*: $\hat{m}_i=\epsilon m_i$, where $\hat{m}_i$ represents the i-th row (all features of a sample) of the perturbed mask matrix, and ϵ represents an independent and identically distributed binary random variable following a Bernoulli distribution. (3) *missing-state-aware patch dropping* (MPD): $m_i=\epsilon m_i,\quad$ where m_i contains 0 (time point is missing), where $\hat{m}_i$ represents the i-th row (a patch) of the perturbed mask matrix, and ϵ represents an independent and identically distributed binary random variable following a Bernoulli distribution, only applied when m_i contains 0 (i.e., the time point is missing). In the above formulas, $\epsilon\sim\text{Bernoulli}(1-\delta)$

holds true.

Among them, DropMissingPatch exhibits the smallest sample variance on time series with the same dropout rate δ, thus providing a more stable training process.

We assume that the downstream task of the MLP model is binary time-series classification, and the output can be represented as:

$$\hat{y}_i = \sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbf{M}_i) + \mathbf{b}) \tag{5}$$

Using cross-entropy as the loss function, the objective function can be represented as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{6}$$

Expanding this, we get:

$$\begin{aligned}\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log\left(\sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbf{M}_i) + \mathbf{b})\right) \\ +(1-y_i)\log\left(1-\sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbf{M}_i) + \mathbf{b})\right)]\end{aligned} \tag{7}$$

where σ represents the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{8}$$

In this way, the mask matrix $\mathbf{M}$ is used in the computation process to ignore missing feature values, thus correctly calculating the objective function even when there are missing values in the input data.

When introducing a random dropout process (such as dropout) and using the perturbed mask matrix $\mathbf{M}'$ instead of the original mask matrix $\mathbf{M}$, we aim to obtain the expected objective function.

First, we define the perturbed mask matrix $\mathbf{M}'$, which represents the result of random dropping at each time step or feature. Assuming the random dropout process is an independent and identically distributed binary random variable, we can represent the expectation of $\mathbf{M}'$ with $\mathbb{E}[\mathbf{M}']$ and the variance of $\mathbf{M}'$ with $\text{Var}(\mathbf{M}')$.

In this case, we can express the processed input features as $(\mathbf{X}_i \odot \mathbf{M}'_i)$ and consider the expected effect of random dropping. The expectation of the objective function can be represented as:

$$\mathbb{E}[\mathcal{L}] = -\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[y_i \log(\hat{y}'_i) + (1-y_i)\log(1-\hat{y}'_i)\right] \tag{9}$$

where $\hat{y}'_i$ is the model output calculated using the perturbed mask matrix $\mathbf{M}'_i$:

$$\hat{y}'_i = \sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbf{M}'_i) + \mathbf{b}) \tag{10}$$

According to the properties of random variables, $\hat{y}'_i$ can be decomposed into its expectation and variance terms:

$$\mathbb{E}[\hat{y}'_i] = \sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbb{E}[\mathbf{M}'_i]) + \mathbf{b}) \tag{11}$$

Assuming $\mathbf{M}'_i$ is a binary random variable with mean $\mathbb{E}[\mathbf{M}'_i]$ and variance $\text{Var}(\mathbf{M}'_i)$. Then we have:

$$\hat{y}'_i = \mathbb{E}[\hat{y}'_i] + \Delta\hat{y}_i \tag{12}$$

where $\Delta\hat{y}_i$ represents the bias caused by the perturbation, and its variance is $\text{Var}(\Delta\hat{y}_i)$.

Let z be a parameter related to the perturbation process, representing the mean of the elements in the perturbation matrix. Assuming the perturbation process ϵ follows a Bernoulli distribution $\text{Bernoulli}(1-\delta)$, then:

$$\mathbb{E}[\epsilon] = 1-\delta \tag{13}$$

Therefore, for each element in the mask matrix, the expectation $\mathbb{E}[m'_{i,j}] = (1-\delta)m_{i,j}$, and further, we can derive:

$$z_i = \sigma(\mathbf{W}(\mathbf{X}_i \odot \mathbb{E}[\mathbf{M}_i]) + \mathbf{b}) \tag{14}$$

In this case, z can be regarded as the expectation of the perturbed matrix elements.

Now, we can express the expected objective function as:

$$\begin{aligned}\mathbb{E}[\mathcal{L}] = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\mathbb{E}[\hat{y}'_i]) + (1-y_i)\log(1-\mathbb{E}[\hat{y}'_i])\right) \\ +\sum_i \frac{1}{2}z_i(1-z_i)\text{Var}(\Delta\hat{y}_i)\end{aligned}$$

where $\sum_i \frac{1}{2}z_i(1-z_i)\text{Var}(\Delta\hat{y}_i)$ is the variance term caused by the perturbation. **As shown in the equation, the random dropout method introduces an additional regularization to the objective function.** For binary classification tasks, this regularization forces the classification probabilities to approach 0 or 1, thereby obtaining a clearer judgment. By reducing the variance of $\Delta\hat{y}_i$, the random dropout method encourages the model to extract more critical high-level representations. Thus, the robustness of the model is enhanced. It should be noted that Equation (3) can be well generalized to multi-class tasks by extending the dimensions of the model output.

In summary, the objective function reflects the impact of the random dropout process on model performance, introducing an additional regularization term that makes the model more robust.

Now, we prove that DropMissingPatch has the smallest sample variance among all existing random dropout methods with the same dropout rate δ.

All random dropout methods for time series are challenged by the instability of the training process. Existing studies have shown that this is due to the introduction of random noise in each training period. These noises increase the difficulty of parameter coverage and the instability of the training process. Generally, sample variance can be used to measure the degree of stability.

As mentioned earlier, all random dropout methods for time series can be transformed into mask operations on the mask matrix M. We can measure the stability of different epochs by comparing the sample variance of the random dropout methods, which can be measured by the Frobenius norm variance $|\mathbf{M}|_F$. To generalize, assume that the original mask matrix $\mathbf{M}$

Models	*Electricity*		*Weather*	
	MSE	MAE	MSE	MAE
TimesNet	0.092	0.210	0.030	0.054
OneFitsAll	0.090	0.207	0.031	0.056
Timer	0.140	0.373	0.056	0.109
INTER (Ours)	**0.070**	**0.188**	**0.028**	**0.054**

Table 4: Imputation performance

is $\mathbf{1}_{n\times n}$, i.e., each element is 1. Therefore, we can calculate its sample variance through the 1-norm of the mask matrix.

The mask matrix is of size $n \times p$, and all random dropout methods can be seen as multITSAe independent Bernoulli samplings. The entire process conforms to a binomial distribution, so we can calculate the variance of $|\mathbf{M}|$.

(1) *Dropout.* n Bernoulli samplings are performed. Dropping an element in the feature matrix leads to d masked elements in the mask matrix. Its sample variance can be calculated by the formula $\mathrm{Var}_{do}(|\mathbf{M}|) = (1-\delta)\delta nd^2$.

(2) *DropPatch.*$\frac{n}{p}$ Bernoulli samplings are performed. Dropping a Patch in the feature matrix leads to pd masked elements in the mask matrix. Its sample variance can be calculated by the formula $\mathrm{Var}_{dp}(|\mathbf{M}|) = (1-\delta)\delta nd^2$.

(3) *MPD.* $\frac{nr}{p}$ Bernoulli samplings are performed. Here, r is the missing probability. Dropping a Patch in the feature matrix leads to pd masked elements in the mask matrix. At the same time, only consider the Patches with missing features. Therefore, its sample variance can be calculated by the formula $\mathrm{Var}_{dmp}(|\mathbf{M}|) = (1-\delta)\delta nrd^2$.

Thus, the variance ranking of the random dropout methods is as follows:

$$\mathrm{Var}_{dmp}(|\mathbf{M}|) \leq \mathrm{Var}_{do}(|\mathbf{M}|) = \mathrm{Var}_{dp}(|\mathbf{M}|) \tag{15}$$

Our *MPD* has the smallest sample variance among all existing random dropout methods. □

D ITSA for Time Series Classification

For incomplete time series classification, we develop an incomplete time series classification loss function based on cross-entropy, which ensures that the model can effectively learn from incomplete data while focusing on the observed components. The loss function is defined as:

$$\mathcal{L}_{\mathbf{cls}} = -\frac{1}{N}\sum_{\mathbf{y}\in\dagger}\left[\frac{\sum_{i=1}^{d}\left(\mathbf{m}_i^T \odot \mathbf{y}_i^T\right)\log(\hat{\mathbf{y}}_i)}{|\mathbf{M}^T|_2^2}\right], \tag{16}$$

where $\odot$ denotes element-wise multITSAication, $\mathbf{y}_i^T$ is the true label of time series $\mathbf{y}_i \in \mathbf{y}$, $\hat{\mathbf{y}}_i$ is the predicted classification, $\mathbf{m}_i^T$ in $\mathbf{M}^T$ is the mask vector for missing values. Finally, the INTER model is trained iteratively to minimize the corresponding objective function.

Models		*SMD*			*MSL*		
		P	R	F1	P	R	F1
Informer 2021	Zero	80.24	68.41	73.85	71.01	70.8	70.9
	BRITS	82.16	70.26	75.75	73.27	74.8	74.03
	SAITS	83.64	71.7	77.21	72.52	73.6	73.06
PatchTST 2022	Zero	78.29	73.5	75.82	75.27	76.24	75.75
	BRITS	79.4	74.62	76.94	77.5	79.16	78.32
	SAITS	80.54	76.63	78.54	73.65	77.25	75.41
Dlinear 2023	Zero	77.87	66.12	71.52	58.37	70.45	63.84
	BRITS	79.14	67.33	72.76	75.04	74.67	74.85
	SAITS	80.67	68.52	74.1	73.83	78.06	75.89
TimesNet 2023	Zero	79.14	76.76	77.93	73.66	75.75	74.69
	BRITS	84.61	78.62	81.51	80.12	81.74	80.92
	SAITS	85.67	79.86	83.13	78.63	80.67	79.64
OneFitsAll 2023	Zero	78.62	65.78	71.63	78.53	79.91	79.21
	BRITS	82.02	69.62	75.31	79.83	81.22	80.52
	SAITS	83.66	69.24	75.77	79.15	81.67	80.39
Timer 2024	Zero	82.24	75.71	78.84	78.22	80.02	79.11
	BRITS	83.25	77.37	80.2	80.2	**82.35**	81.26
	SAITS	85.08	79.45	82.17	80.04	81.29	80.66
TriD-MAE	\	75.94	64.92	70	75.09	75.89	75.49
INTER (Ours)	\	**86.33**	**81.2**	**83.69**	**80.24**	82.33	**81.27**

Table 5: Anomaly detection performance comparison under different datasets. The P, R, and F1 represent the precision, recall, and F1-score (%), respectively.

E Additional Results

E.1 Incomplete Time Series Imputation and Anomaly Detection

Incomplete Time Series Imputation

In this paper, we select the datasets from the electricity and weather scenarios as our benchmarks, including *ETT* (Zhou et al, 2021), *Electricity* (UCI), and *Weather* (Wetterstation), where the data-missing problem happens commonly. To compare the imputation model capacity under different proportions of missing data, we randomly mask the time points in the ratio of 12.5%, 25%, 37.5%, 50%. Table 5 presents a comparative analysis of imputation performance across two datasets. INTER achieves the best performance in every case Furthermore, INTER demonstrates the most stable imputation accuracy, further validating its effectiveness.

Incomplete Time Series Anomaly Detection

Table 5 compares the anomaly detection performance on the SMD and MSL datasets using precision (P), recall (R), and F1-score (F1) as evaluation metrics. Our method achieves superior results, obtaining the highest F1-scores of 83.69% on SMD and 81.27% on MSL. These findings highlight the robustness of our approach in accurately detecting anomalies, even in complex scenarios. Furthermore, the model consistently balances precision and recall, ensuring reliable and comprehensive detection performance across both datasets.

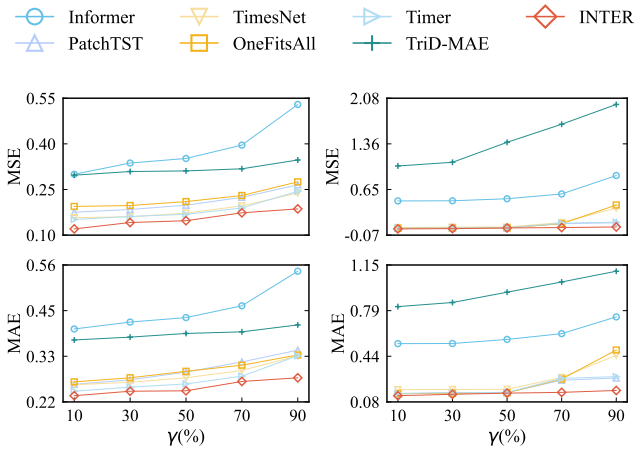
E.2 Ablation Study

We analyze the impact of various components of INTER on prediction performance, as shown in Table 6. Several variants

Models	MSE	MAE
INTER	**0.049**	**0.158**
w/o MPL	0.050	0.160
w/o ITSA	0.053	0.162
w/ DropPatch (Threshold)	0.052	0.162
w/ DropPatch (Fully Random)	0.058	0.173
w/ DropPoint	0.050	0.160

Table 6: Ablation Study

of INTER are tested to evaluate the effectiveness of its components: INTER without MPL (w/o MPL), INTER without ITSA (w/o ITSA), and variants with DropPatch. The baseline model INTER achieves an MSE of 0.049 and an MAE of 0.158. Removing MPL increases both MSE (0.050) and MAE (0.160), demonstrating its contribution to model accuracy. Similarly, excluding ITSA results in a further increase in error (MSE 0.053, MAE 0.162), indicating the importance of this component. The DropPatch with Threshold variant (w/ DropPatch(Threshold)) causes a minor increase in error (MSE 0.052, MAE 0.162), while the Fully Random variant (w/ DropPatch(Fully Random)) leads to a more significant performance drop (MSE 0.058, MAE 0.173). Notably, DropPoint performs comparably to the baseline model (MSE 0.050, MAE 0.160). These results underscore the effectiveness of MPL and ITSA in enhancing model performance and demonstrate that structured methods, like DropPatch with Threshold, outperform fully random approaches.



(a) *Electricity* (b) *Exchange*

Figure 4: The forecasting performance vs. missing rate

E.3 Effect of Missing Rate

When varying the missing rate γ (i.e., how many features/values in multivariate time series data are dropped) from 10% to 90%, the corresponding experimental results w.r.t. the time series forecasting task over *Electricity* and *Weather* datasets are depicted in Figure 4. We can find that, with the growth of the missing rate, the forecasting accuracy (i.e., MSE and MAE) of each algorithm descends consistently. It is attributed to the less data information for forecasting when the missing rate turns high. Among these algorithms, INTER performs the best in each case. Moreover, its accuracy becomes more stable with the increase in missing rate. In other words, INTER is more robust with the increasing missing rate γ than others.