Supplementary Materials
SAND: Smooth Imputation Of Sparse And Noisy Functional Data With Transformer Networks

Ju-Sheng Hong[*]
Department of Statistics
University of California Davis
Davis, CA 95616
jsdhong@ucdavis.edu

Junwen Yao
Department of Statistics
University of California Davis
Davis, CA
jwyao@ucdavis.edu

Jonas Mueller
Cleanlab
Cambridge, MA
jonaswmueller@gmail.com

Jane-Ling Wang
Department of Statistics
University of California Davis
Davis, CA
janelwang@ucdavis.edu

We present the proof of our major results in section 1 and additional numerical experiments in section 2. In addition to the notations introduced in the main text, we add a few more in the supplement. The bold symbol $\mathbf{x}_a$ represents a length-a vector (either a row or column vector) with a constant value x. In a similar fashion, we use $\mathbf{x}_{a \times b}$ to denote an $a \times b$ matrix with a constant value x.

1 Missing proofs

1.1 Proof of Theorem 1

One can easily see that $\mathcal{I}$ is dense in $[0, 1]$ as M approaches infinity. Let $x(t)$ be an Riemann integrable function defined on $[0, 1]$ and x_M be the vector whose entries are x evaluated at points in $\mathcal{I}$, i.e.,

$$x_M = \left(x(0), x\left(\frac{1}{M-1}\right), x\left(\frac{2}{M-1}\right), \ldots, x(1) \right).$$

For any positive number $a \in [0, 1]$, define $a_M = \min\{z \mid a \leq z/(M-1), z \in \mathbb{Z}^+\}$. Since $a_M/(M-1) \to a$ as $M \to \infty$, we can verify that

$$\lim_{M \to \infty} \left[\texttt{cumsum}\left(\frac{x_M}{M-1}\right) \right]_{a_M} = \lim_{M \to \infty} \frac{1}{M-1} \sum_{i=1}^{a_M} x\left(\frac{i}{M-1}\right) \tag{1}$$

$$= \int_0^a x(t)\, dt, \tag{2}$$

where (1) is the upper Riemann sum of $x(\cdot)$ on $[0, a_M]$ and it becomes (2) since $x(\cdot)$ is Riemann integrable. Hence, the cumsum operator becomes an integral operator as $M \to \infty$. To complete the proof, we remark that the integral of a continuous function is differentiable; thus, the imputation from SAND is differentiable provided that $\widetilde{T}$ is continuous. $\qquad \square$

---

[*]Correspondence to: Ju-Sheng Hong ¡jsdhong@ucdavis.edu¿.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1.2 Proof of Theorem 2

Recall the definitions of $\widetilde{T}$, $\widetilde{T}_c$, $\widetilde{\mathbf{T}}$, and $\widetilde{\mathbf{T}}_c$ in section 3.2. By Lemma 1.1, for any h_d, there exists a SAND and a corresponding M-by-M matrix $\mathbf{P}$ such that the imputation is $\widehat{T}_i = (\widetilde{T}_i)_1 + \widetilde{T}_{i,c}\mathbf{P}$ for subject i. Therefore,

$$\|d_i^{\text{mask}} \odot (\widetilde{T}_i - \widehat{T}_i)\|_2^2 \;=\; \|d_i^{\text{mask}} \odot [\widetilde{T}_i - ((\widetilde{T}_i)_1 + \widetilde{T}_{i,c}\mathbf{P})]\|_2^2$$

$$=\; \|d_i^{\text{mask}} \odot (\widetilde{T}_{i,c} - \widetilde{T}_{i,c}\mathbf{P})\|_2^2,$$

where $u \odot v$ is the element-wise product of u and v. To finish this proof, we first prove the inequality in Theorem 2 by giving an explicit form of $\mathbf{P}$ and then show that this particular $\mathbf{P}$ can be learned by SAND. Without loss of generality, we can assume that the first element in $\widetilde{T}_i$ is 0 for all subjects. Since $d_i^{\text{mask}} = (d_{i1}, \ldots, d_{iM})$ and $d_{ij} \in \{0, 1\}$ for all $i \in \mathbb{A}$ and $j \in \{1, \ldots, M\}$, we have

$$\sum_{i \in \mathbb{A}} \|d_i^{\text{mask}} \odot (\widetilde{T}_i - \widetilde{T}_i\mathbf{P})\|_2^2 \;\leq\; \sum_{i \in \mathbb{A}} \|\widetilde{T}_i - \widetilde{T}_i\mathbf{P}\|_2^2 \tag{3}$$

$$=\; \sum_{i \in \mathbb{A}} \widetilde{T}_i \mathbf{P}\mathbf{P}^{\intercal}\widetilde{T}_i^{\intercal} - \sum_{i \in \mathbb{A}} \widetilde{T}_i \mathbf{P}\widetilde{T}_i^{\intercal} - \sum_{i \in \mathbb{A}} \widetilde{T}_i \mathbf{P}^{\intercal}\widetilde{T}_i^{\intercal} + \sum_{i \in \mathbb{A}} \widetilde{T}_i \widetilde{T}_i^{\intercal}. \tag{4}$$

To simplify our argument, we denote $\mathbf{X}_T$ as $\sum_i \widetilde{T}_i^{\intercal}\widetilde{T}_i$ and let $\text{tr}(\cdot)$ be the trace operator. Since $\text{tr}(\cdot)$ is invariance under cyclic permutation, (4) becomes

$$\text{tr}(\mathbf{P}\mathbf{P}^{\intercal}\mathbf{X}_T) - \text{tr}(\mathbf{P}\mathbf{X}_T) - \text{tr}(\mathbf{P}^{\intercal}\mathbf{X}_T) + \text{tr}(\mathbf{X}_T). \tag{5}$$

Suppose that $\mathbf{P}$ is a symmetric matrix and $\mathbf{P} = \mathbf{U}_P\Sigma_P\mathbf{U}_P^{\intercal}$ is the eigendecomposition, where $\Sigma_P = \text{diag}(a_1, \ldots, a_M)$ and $\{a_j\}_{j=1}^M$ is the non-increasing eigenvalues. Then,

$$(5) \;=\; \text{tr}(\mathbf{U}_P\Sigma_P^2\mathbf{U}_P^{\intercal}\mathbf{X}_T) - 2\text{tr}(\mathbf{U}_P\Sigma_P\mathbf{U}_P^{\intercal}\mathbf{X}_T) + \text{tr}(\mathbf{X}_T). \tag{6}$$

As $\mathbf{X}_T$ is symmetric, it has the eigendecomposition, namely $\mathbf{X}_T = \mathbf{U}_T\Sigma\mathbf{U}_T^{\intercal}$ where $\Sigma = \text{diag}(\lambda_1, \ldots, \lambda_M)$ and $\{\lambda_j\}_{j=1}^M$ is the non-increasing eigenvalues. We pick $\mathbf{U}_P = \mathbf{U}_T$ and denote the diagonal elements of $\mathbf{U}_T^{\intercal}\mathbf{X}_T\mathbf{U}_T$ as $a_1, \ldots, a_M$. Using the property of trace where it is invariant under cyclic permutation again, (6) becomes

$$\sum_{j=1}^M a_j^2\lambda_j - 2a_j\lambda_j + \lambda_j. \tag{7}$$

Next, we want to minimize (7) over $a_j \in \mathbb{R}$ for all j. First, we notice that $\lambda_j \geq 0$ for all j since $\mathbf{X}_T = \mathbf{U}_T\Sigma\mathbf{U}_T^{\intercal}$ is a semi-positive definite matrix. Then, we set the first derivative of (7) with respect to a_j as 0 and get $a_j = 1$ for all j. However, when $\mathbf{P}$ is not full-rank, say $\text{rank}(\mathbf{P}) = d_\epsilon$, there will be $(M - d_\epsilon)$ zeros in $\{a_j\}_{j=1}^M$. As a result, all a_j must be in $\{0, 1\}$ regardless of the rank of $\mathbf{P}$. Under the rank insufficient case, we need to decide which a_j should be 0. To do this, we notice that

$$a_j^2\lambda_j - 2a_j\lambda_j + \lambda_j \;=\; \begin{cases} 0, & \text{when } a_j = 1; \\ \lambda_j, & \text{when } a_j = 0. \end{cases} \tag{8}$$

Consequently, if $\mathbf{P}$ is of rank d_ϵ, $a_1, \ldots, a_{d_\epsilon}$ must be 1 to minimize (7), as $\{\lambda_j\}_{j=1}^M$ is non-increasing. To sum up, the minimum of (3) over $\mathbf{P}$ being symmetric with rank d_ϵ is obtained when

$$\mathbf{P} \;=\; \mathbf{U}_T\text{diag}(\mathbf{1}_{d_\epsilon}, \mathbf{0}_{M-d_\epsilon})\mathbf{U}_T^{\intercal}. \tag{9}$$

In addition, (8) implies that the minimum of (3) is $\sum_{j=d_\epsilon+1}^M \lambda_j$. As d_ϵ satisfies $\sum_{j=1}^d \lambda_j / \sum_{j=1}^M \lambda_j \geq 1 - \epsilon$ and $\sum_{i \in \mathbb{A}} \|\widetilde{T}_i\|_2^2 = \sum_{i \in \mathbb{A}} \widetilde{T}_i\widetilde{T}_i^{\intercal} = \text{tr}(\mathbf{X}_T) = \sum_{j=1}^M \lambda_j$, we have

$$\frac{1}{\|\mathbb{A}\|_0} \sum_{i \in \mathbb{A}} \|d_i^{\text{mask}} \odot (\widetilde{T}_i - \widehat{T}_i)\|_2^2 \;\leq\; \frac{\epsilon}{\|\mathbb{A}\|_0} \sum_{i \in \mathbb{A}} \|\widetilde{T}_i\|_2^2$$

provided that $\widehat{T}_i$ can be written as $\widetilde{T}_i\mathbf{P}$ where $\mathbf{P}$, described in (9), can be learned by a SAND.

To show that there exists a SAND which learns $\mathbf{P}$, we let

2

- $\mathbf{U}_{T,\epsilon}$ denote the truncated $\mathbf{U}_T$ where only the first d_ϵ columns are preserved, i.e., $\mathbf{U}_{T,\epsilon} = \mathbf{U}_T[:, 1, \ldots, d_\epsilon]$,
- $\mathbf{I}$ be the positional encoding matrix for the output grid $\mathcal{I}$,
- $\mathbf{B}$ as an M-by-M matrix where the elements of $\mathbf{B}$ satisfy

$$(\mathbf{B})_{ij} = \begin{cases} 1/(M-1), & \text{if } j \geq i; \\ 0, & \text{otherwise.} \end{cases}$$

Then by Lemma 1.1 combined with (11) and (12), an imputation from such SAND becomes

$$\widetilde{T}_i \left[\left(\mathbf{W}_K \widetilde{\mathbf{T}}_c \right)^\top \left(\mathbf{W}_Q \widetilde{\mathbf{T}}_c \right) \Big/ \sqrt{h_d} \right] \mathbf{B} \tag{10}$$

and by taking $\mathbf{W}_K = \sqrt{h_d} \mathbf{U}_{T,\epsilon}^\top (\mathbf{0}\ \mathbf{I}^{-1})$ and $\mathbf{W}_Q = \sqrt{h_d} \mathbf{U}_{T,\epsilon}^\top \mathbf{B}^{-1} (\mathbf{0}\ \mathbf{I}^{-1})$, we have (10) $=$ $\widetilde{T}_i \mathbf{U}_{T,\epsilon} \mathbf{U}_{T,\epsilon}^\top = \widetilde{T}_i \mathbf{P}$. $\qquad\square$

1.3 Supporting Lemmas for Theorem 2

Lemma 1.1. *Suppose that SAND has a single-head structure. Then for any h_d, there exists a SAND with a corresponding matrix $\mathbf{P}$ such that, for any subject i, the imputation can be written as $(\widetilde{T}_i)_1 + \widetilde{T}_{i,c} \mathbf{P}$, and it becomes $\widetilde{T}_i \mathbf{P}$ when $(\widetilde{T}_i)_1 = 0$.*

Proof. Recall that

$$\text{SAND}(\widetilde{\mathbf{T}}_i) = (\widetilde{T}_i)_1 + \text{Intg}(\widetilde{D}_i)$$

$$= (\widetilde{T}_i)_1 + \text{Intg}[\text{Diff}(\widetilde{\mathbf{T}}_{i,c})],$$

where $\text{Diff}(\cdot)$ and $\text{Intg}(\cdot)$ are defined according to

$$\widetilde{D}_i = \text{Diff}(\widetilde{\mathbf{T}}_{i,c}) = W_O \left(\mathbf{W}_V \widetilde{\mathbf{T}}_{i,c} \right) \left[\left(\mathbf{W}_K \widetilde{\mathbf{T}}_{i,c} \right)^\top \left(\mathbf{W}_Q \widetilde{\mathbf{T}}_{i,c} \right) \Big/ \sqrt{h_d} \right],$$

$$\text{Intg}(\widetilde{D}_i) = \texttt{cumsum}\left[\widetilde{D}_i/(M-1) \right].$$

Define $\mathbf{B}$ as an M-by-M matrix where the elements of $\mathbf{B}$ satisfy

$$(\mathbf{B})_{ij} = \begin{cases} 1/(M-1), & \text{if } j \geq i; \\ 0, & \text{otherwise.} \end{cases}$$

Then the output of SAND can be written as

$$\text{SAND}(\widetilde{\mathbf{T}}_i) = (\widetilde{T}_i)_1 + \text{Intg}[\text{Diff}(\widetilde{\mathbf{T}}_{i,c})]$$

$$= (\widetilde{T}_i)_1 + W_O \left(\mathbf{W}_V \widetilde{\mathbf{T}}_{i,c} \right) \left[\left(\mathbf{W}_K \widetilde{\mathbf{T}}_{i,c} \right)^\top \left(\mathbf{W}_Q \widetilde{\mathbf{T}}_{i,c} \right) \Big/ \sqrt{h_d} \right] \mathbf{B}. \tag{11}$$

Define

$$\mathbf{P}_i = \left[\left(\mathbf{W}_K \widetilde{\mathbf{T}}_{i,c} \right)^\top \left(\mathbf{W}_Q \widetilde{\mathbf{T}}_{i,c} \right) \Big/ \sqrt{h_d} \right] \mathbf{B}.$$

Because $\mathbf{P}_i$ is an M-by-M matrix, by taking

$$W_O = (1\ \ \mathbf{0}_{1\times d}) \quad \text{and} \quad \mathbf{W}_V = \begin{bmatrix} \mathbf{1}_{1\times M} \\ \mathbf{0}_{d\times M} \end{bmatrix}, \tag{12}$$

we obtain $W_O(\mathbf{W}_V \widetilde{\mathbf{T}}_{i,c}) = \widetilde{T}_{i,c}$. Also it is worth noticing that $W_O(\mathbf{W}_V \widetilde{\mathbf{T}}_{i,c}) = \widetilde{T}_i$ when $(\widetilde{T}_i)_1 = 0$. To show that there exists a matrix $\mathbf{P}$ such that $\mathbf{P}_i \equiv \mathbf{P}$ for all i, we denote $\mathbf{I}$ as the positional encoding of the output grid $\mathcal{I}$. For any subject i, we have $\widetilde{\mathbf{T}}_{i,c} = \begin{bmatrix} \widetilde{T}_{i,c} \\ \mathbf{I} \end{bmatrix}$. Therefore, when the first columns in $\mathbf{W}_K$ and $\mathbf{W}_Q$ are zeros, all $\mathbf{P}_i$ are the same. $\qquad\square$

3

1.4 Proof of Corollary 1

Lemma 1.1 shows that there exists a $\mathbf{P}$ such that.

$$\|d_i^{\mathrm{mask}} \odot (\widetilde{T}_i - \widehat{T}_i)\|_2^2 \;=\; \|d_i^{\mathrm{mask}} \odot [\widetilde{T}_i - ((\widetilde{T}_i)_1 + \widetilde{T}_{i,c}\mathbf{P})]\|_2^2$$

$$=\; \|d_i^{\mathrm{mask}} \odot (\widetilde{T}_{i,c} - \widetilde{T}_{i,c}\mathbf{P})\|_2^2.$$

When all elements in $d_i^{\mathrm{mask}} = 1$, the last expression becomes

$$\|(\widetilde{T}_{i,c}(I_M - \mathbf{P})\|_2^2.$$

Notice that the rank of $\mathbf{P}$ is determined by the minimum of $\mathrm{rank}(\mathbf{W}_K)$, $\mathrm{rank}(\mathbf{W}_Q)$ and $\mathrm{rank}(\widetilde{\mathbf{T}}_c)$. As $d = M$, $\mathrm{rank}(\widetilde{\mathbf{T}}_c) = M$ and $\mathrm{rank}(\mathbf{P}) = h_d = d_\epsilon$. To see the minimum is attained at $\epsilon/\|\mathbb{A}\|_0 \sum_{i \in \mathbb{A}} \|\widetilde{T}_{i,c}\|_2^2$, we follow the proof of Theorem 2 and notice that in the settings of this corollary, the inequality in (3) is actually an equality. Thus, the minimum is $\frac{\epsilon}{\|\mathbb{A}\|_0} \sum_{i \in \mathbb{A}} \|\widetilde{T}_{i,c}\|_2^2$. □

2 Additional experiment study

Code availability & reproducibility. The code, checkpoints, and real datasets are provided in a separate `.zip` file.

Numerical analysis of Diff. In Figure S1, we illustrate the effectiveness of Diff in capturing the overall trend of the true derivative (the black dashed curve) using simulated data where observations (blue dots) are sampled from the underlying process (the green dashed curve). Despite lacking direct access to the derivative and no explicit instruction to do so, Intg indirectly guides Diff to produce outcomes (the orange curve) akin to derivatives. As illustrated in the right panel of Figure S1, Diff adeptly captures the general trend of the true derivative, despite operating without this information.
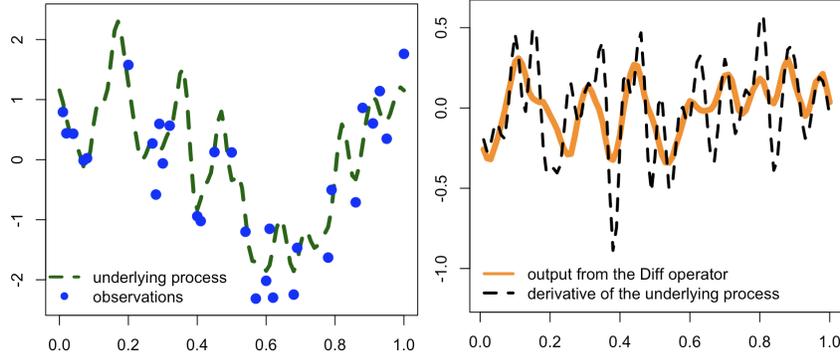
Figure S1: The comparison between a Diff output (the orange curve) and the first derivative (the black curve) of the corresponding underlying process (the green dashed curve).

2.1 Discussion of [3, 1, 2]'s methods

In this section, we provide additional details regarding the comparison of our method with [3, 1]'s methods. [3] relies on the EM algorithm (computationally expensive), known to be unstable with sparse data, as highlighted in [1]. [1] is later improved by [2] as disclosed in James' website so we only include [2]'s method in our analyses. [2]'s package is called `FPCA` in the `R` software; however, to prevent confusion with the established functional principal component analysis (FPCA), we denote their method as mFPCA. Here, the 'm' signifies its maximum likelihood foundation, distinguishing it as a likelihood-based approach.

2.2 Additional Simulation Results

Our experiments involve six scenarios which differ in terms of the distribution of the scores (a_{ik} and b_{ik}) and the complexity (i.e. number of eigen-components K) of the underlying data-generating processes. Specifically, we consider three different generating distributions:

4

- $\mathcal{E}$: the exponential distribution with rate 1, which reflects a heavy-tailed process;

- $\mathcal{T}$: the t-distribution with 5 degrees of freedom;

- $\mathcal{G}$: the standard Gaussian distribution.

Beyond different generating distributions, we also study how the data dimension K affects various imputation methods, considering a:

- low dimensional case (`LowDim`) where: $K = 5$

- high dimensional case (`HighDim`) where: $K = 20$

Table S1 categorizes the experiment outcomes according to their setup and provides links to the tables containing the details of each experiment. Except for Table 1 in the main text, the remaining tables are presented in this supplement (indicated by the prefix S).

Table S1: Correspondence between experimental setups and tables of results.

	HighDim			LowDim		
	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{G}$
Tables	1, S2, S3, S4 & S5	S7	S9	S6	S8	S10

Result. In Table S2, we look at the same scenario as in Table 1 and use transformers with ReLU activation (RT) to report the Mean Squared Errors (MSE) and Total Variations (TV). We find that RT and GT (GeLU-activated transformers) perform similarly. Figure 1 reveals that both RT and GT models face a problem with smoothness in their imputations. Next, we assume the signal-to-noise ratio is 0 and investigate the performance of SAND under the error-free case. In Table S3, there is an overall decrease in MSE and TV among most methods, and SAND demonstrates the best performance. Given the similar performance between RT and GT, we focus on GT results for subsequent scenarios. In Tables S4 & S5, we deal with a scenario similar to Tables S2 & S3, except that the time points t_{ij} are now dependent within each subject i. In this scenario, Transformer-based models outperform all other basic methods. Among transformers, SAND has the smallest MSE in every case and it effectively reduces TV, demonstrating its ability to create smooth imputations. Tables S6, S7, S8, S9, S10 report the performance of eight imputation methods: PACE, FACE, mFPCA, MICE, CNP, GAIN, 1DS and SAND. Among these methods, either PACE or SAND consistently outperforms the other six methods.

When the underlying process is a Gaussian process, Tables S9 and S10 show that PACE performs the best (as expected in this setting where its assumptions are perfectly met). This can be explained partly by equation $E(\xi_{ik} \mid \widetilde{Y}_i) = \lambda_k \phi_{ik}^{\mathsf{T}} \Sigma_{Y_i}^{-1} (Y_i - \mu_i)$, which holds under the Gaussian assumption. While SAND does not reach the same level of performance as PACE for Gaussian scores, it still significantly outperforms the other methods. Interestingly, the performance gap between PACE and SAND narrows as the number of basis functions used in the data-generating process increases.

For all other scenarios with asymmetric distributions, Tables S2, S3, S4, S5 and S6 illustrate that SAND takes the lead over PACE in all situations, except for very simple scenarios where the data is densely observed and the number of basis functions is low.

For score distributions following a t-distribution, as shown in Tables S7 and S8, SAND tends to deliver better results than PACE as the number of basis functions used in the data-generating process increases. In general, when the scores are non-Gaussian and the number of basis functions used in the data-generating process is high, SAND consistently outperforms PACE irrespective of the sparsity of observations, as depicted in Tables S2, S3, S4, S5 and S7.

In summary, while PACE excels in handling simple scenarios like Gaussian processes with a low number of basis functions, SAND offers more robust imputation performance across a broader range of data-generating processes, including asymmetric or t-distributions, and those involving a large number of basis functions.

5

2.3 Limitations of SAND

While SAND demonstrates superior performance across various complex scenarios and asymmetric score distributions, it has limitations when applied to simpler data structures, particularly when the underlying process is generated by a low number of basis functions *and* the score distribution is symmetric. In these cases, SAND does not outperform PACE, which is specifically designed to excel under symmetric distribution assumptions.

PACE's exceptional performance in these simpler scenarios is due to the fact that the setting perfectly fits its model assumptions. This gives PACE a distinct advantage, making it the optimal choice under such conditions. However, despite this specific advantage of PACE, SAND still maintains competitive performance, often ranking as the second-best method among ten competing techniques. In addition, as the number of basis functions increases, the performance gap between PACE and SAND narrows. The Mean Squared Errors (MSE) of SAND are comparable to those of PACE, indicating that while PACE may have a slight edge, SAND remains a viable method.

The primary reason for SAND's comparatively lower performance in the non-complex scenarios lies in its neural network architecture. Neural networks generally exhibit their strengths when handling complex data structures. Therefore, when faced with simpler data, SAND's advantages are less pronounced.

In summary, SAND's limitations are most evident in scenarios where both the data-generating process is simple *and* the score distribution is symmetric that are ideally suited for PACE. In such cases, PACE outperforms SAND. However, SAND still delivers strong results, showcasing its robustness and flexibility across a broad range of scenarios.

2.4 Additional Data Application Results

In this section, we alter our approach by sampling time points t_{ij} in a dependent manner within each subject in the UK electricity dataset and present the results in Table S11. Once again, SAND maintains its superior performance in terms of MSE. A direct comparison between rows representing by GT and SAND suggests its ability to generate smooth imputations. This observation aligns with the evidence presented in Figure S2.
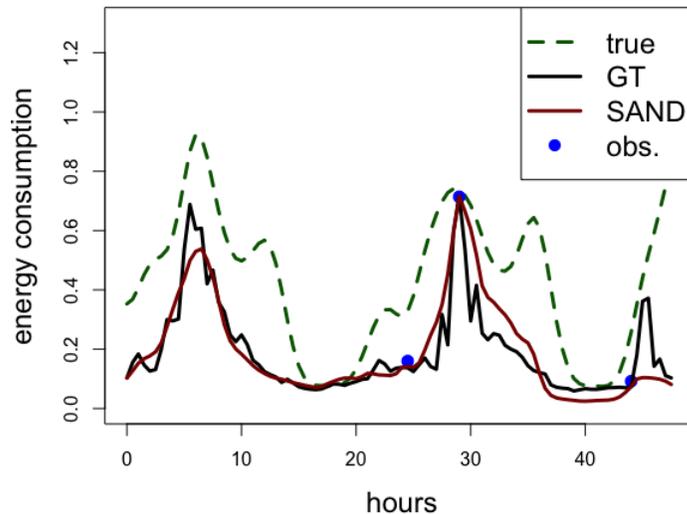
Figure S2: An imputation by SAND from the UK electricity testing set. The green curve represents the underlying electricity usage where three observations (blue dots) are sampled. The black line ('GT') refers to the imputation from a vanilla transformer and the red line is the output of SAND.

6

Table S2: Full Tables 1 from the main paper. The eigenvalues follow an exponential with rate 1, the number of bases is 40 ($K = 20$), the time points within any subject are independently sampled, and the signal-to-noise ratio is 4. Bold values indicate the top 2 performing methods.

	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	189.9(4.3)	187.1(2.0)	450.0(15)	201.9(2.1)	795.5(33)	209.5(2.2)
FACE	284.6(8.8)	198.9(2.1)	488.2(16)	204.5(2.2)	807.1(32)	209.5(2.2)
mFPCA	224.7(5.8)	192.0(2.1)	480.3(16)	204.0(2.2)	787.1(31)	**209.3**(2.2)
MICE	176.7(3.7)	233.1(1.7)	721.6(27)	318.4(3.0)	1416(57)	332.7(2.8)
CNP	290.4(11)	198.9(2.0)	551.3(21)	207.6(2.1)	920.3(52)	211.9(2.2)
GAIN	261.9(6.8)	350.0(3.4)	1454(52)	413.1(5.1)	1862(51)	385.4(4.3)
1DS	262.9(6.0)	273.8(2.4)	735.3(22)	305.7(3.7)	1157(43)	263.3(3.1)
ReLU-activated transformers with penalties						
RT1	172.1(3.2)	213.2(1.8)	439.2(15)	225.0(2.2)	802.1(35)	232.0(2.5)
RT1P	170.6(3.4)	192.6(1.9)	426.5(14)	202.9(2.1)	781.5(34)	212.9(2.3)
RT1S	172.0(3.2)	213.2(1.8)	439.1(15)	225.0(2.2)	801.3(35)	232.0(2.5)
RT2	172.6(3.4)	225.2(1.7)	432.4(14)	216.1(2.3)	801.7(39)	224.8(2.2)
RT2P	174.2(3.9)	181.9(2.0)	425.1(13)	198.7(2.3)	791.2(39)	213.9(1.9)
RT2S	165.3(3.2)	177.3(1.6)	429.9(13)	206.7(2.3)	801.7(39)	224.8(2.2)
GeLU-activated transformers with penalties						
GT1	169.8(3.2)	218.2(1.7)	436.7(15)	227.0(2.2)	798.6(35)	230.6(2.6)
GT1P	169.0(3.5)	179.9(2.0)	425.3(14)	**199.4**(2.1)	**777.4**(34)	210.2(2.2)
GT1S	169.8(3.2)	218.1(1.7)	436.7(15)	227.0(2.2)	796.5(35)	227.6(2.6)
GT2	174.8(3.5)	223.0(1.7)	433.8(14)	221.9(2.1)	804.5(39)	226.1(2.4)
GT2P	179.9(3.9)	182.1(2.0)	**422.6**(13)	**199.4**(2.1)	788.9(39)	210.0(2.2)
GT2S	**160.8**(3.4)	**168.5**(1.5)	427.7(13)	208.8(2.1)	804.5(39)	226.1(2.4)
GeLU-activated transformers with augmented modules						
ATT	185.1(3.8)	220.0(1.7)	446.9(14)	220.6(2.1)	852.0(42)	224.0(2.5)
SAND	**146.5**(2.7)	**164.6**(1.8)	**410.9**(13)	**196.8**(2.0)	**758.1**(43)	**206.8**(2.2)
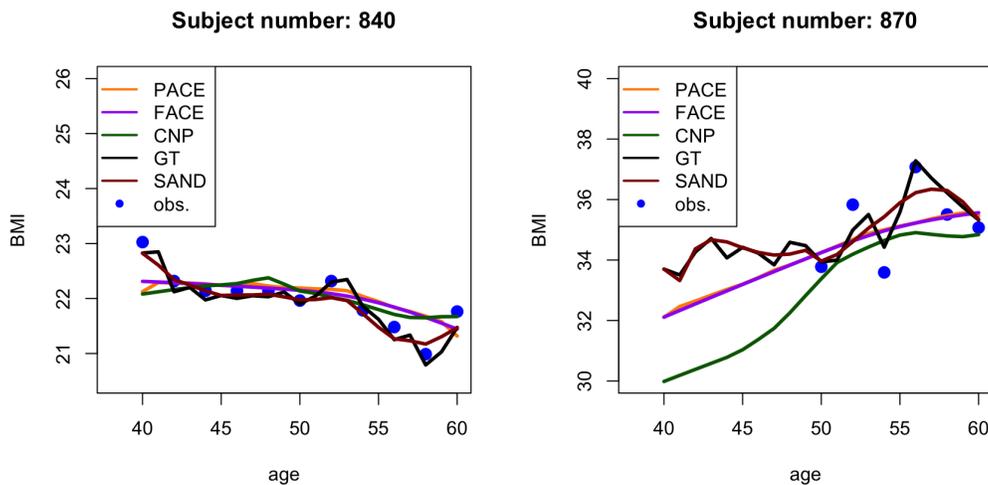
Figure S3: Imputations from the top 4 performing methods in the Framingham Heart Study dataset.

7

Table S3: The setting is similar to the one used in Table S2: the eigenvalues follow an exponential with rate 1, the number of bases is 40 ($K = 20$) and the time points within any subject are independently sampled; except that the signal-to-noise ratio is 0. Bold values indicate the top 2 performing methods.

	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	155.8(4.3)	183.8(2.1)	400.4(14)	200.3(2.2)	744.4(31)	208.9(2.2)
FACE	264.4(8.8)	198.0(2.1)	449.6(16)	203.6(2.2)	757.5(30)	208.9(2.2)
mFPCA	196.8(5.7)	190.0(2.1)	438.4(15)	203.2(2.2)	860.1(43)	211.6(2.2)
MICE	112.4(3.6)	145.2(1.6)	689.7(26)	299.4(3.0)	1403(57)	323.2(2.8)
CNP	261.2(10)	197.0(2.1)	542.0(27)	206.9(2.2)	845.7(43)	210.9(2.2)
GAIN	169.9(4.4)	229.8(2.4)	1179(43)	383.0(3.9)	1951(54)	423.7(2.6)
1DS	157.4(5.4)	193.1(2.2)	648.6(21)	283.6(3.7)	1082(43)	253.6(2.8)
ReLU-activated transformers with penalties						
RT1	**96.22**(2.3)	152.8(1.6)	373.8(14)	209.9(2.2)	730.5(35)	225.7(2.5)
RT1P	125.5(2.9)	179.6(2.0)	374.4(13)	199.0(2.3)	**714.1**(34)	207.2(2.2)
RT1S	97.11(2.4)	**151.2**(1.5)	372.3(15)	196.1(2.1)	726.0(34)	219.3(2.3)
RT2	99.79(2.4)	151.5(1.6)	373.5(14)	221.4(2.2)	733.8(37)	223.5(2.5)
RT2P	132.3(3.0)	183.3(1.9)	370.2(14)	204.6(2.1)	718.2(35)	210.1(2.1)
RT2S	102.5(2.4)	156.6(1.6)	368.4(13)	213.3(2.1)	736.7(37)	225.9(2.5)
GeLU-activated transformers with penalties						
GT1	98.22(2.3)	153.3(1.6)	371.8(13)	207.6(2.2)	735.9(35)	226.4(2.4)
GT1P	131.5(3.0)	182.7(2.0)	372.5(13)	197.1(2.1)	719.5(34)	208.8(2.2)
GT1S	101.0(2.4)	156.8(1.5)	**368.2**(13)	198.2(2.1)	731.0(35)	219.0(2.3)
GT2	100.9(2.4)	152.4(1.6)	376.1(14)	211.2(2.2)	736.8(37)	224.8(2.5)
GT2P	133.0(3.0)	183.0(2.0)	372.4(13)	**196.7**(2.1)	720.1(36)	**208.8**(2.2)
GT2S	105.7(2.5)	156.1(1.6)	371.7(13)	200.2(2.1)	736.8(37)	224.8(2.5)
GeLU-activated transformers with augmented modules						
ATT	122.6(3.2)	170.6(1.7)	416.3(13)	211.4(2.2)	808.4(34)	221.1(2.4)
SAND	**90.88**(2.3)	**147.7**(1.6)	**347.3**(13)	**190.8**(2.1)	**688.4**(32)	**208.2**(2.2)

Table S4: The eigenvalues follow an exponential with rate 1, the number of bases is 40 ($K = 20$), the time points within any subject are dependently sampled, and the signal-to-noise ratio is 4. Bold values indicate the top 2 performing methods.

	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	325.9(10)	193.7(2.1)	567.1(19)	204.5(2.2)	839.3(37)	210.6(2.2)
FACE	419.7(15)	203.1(2.2)	659.5(24)	207.0(2.2)	842.7(34)	210.4(2.2)
mFPCA	354.6(11)	199.4(2.1)	588.9(21)	206.2(2.2)	859.2(34)	212.0(2.2)
MICE	395.1(17)	235.4(1.7)	815.3(29)	276.9(2.5)	1401(56)	321.3(2.8)
CNP	709.2(35)	211.2(2.2)	696.3(33)	210.2(2.2)	1007(45)	212.4(2.2)
GAIN	676.7(31)	379.1(3.9)	1564(57)	431.1(4.3)	1955(56)	420.4(2.7)
1DS	567.8(21)	266.2(2.1)	904.1(32)	276.0(3.1)	1157(42)	255.0(2.8)
GeLU-activated transformers with penalties						
GT1	307.2(9.2)	246.9(1.8)	575.7(18)	250.8(2.5)	842.1(40)	255.8(2.4)
GT1P	304.0(9.2)	189.2(2.0)	553.8(18)	**203.1**(2.1)	**828.2**(40)	**209.3**(2.4)
GT1S	**286.1**(9.1)	**179.7**(1.7)	575.7(18)	250.8(2.5)	841.7(40)	255.8(2.3)
GT2	324.3(11)	248.8(1.8)	573.7(20)	254.9(2.5)	845.6(43)	259.2(2.1)
GT2P	319.7(10)	190.2(2.0)	**551.0**(19)	203.4(2.2)	830.1(43)	210.3(2.0)
GT2S	302.0(11)	181.0(1.7)	573.7(20)	254.9(2.5)	845.6(43)	259.2(2.1)
GeLU-activated transformers with augmented modules						
ATT	307.2(9.8)	224.1(1.8)	583.3(18)	228.5(2.4)	902.3(45)	227.0(2.4)
SAND	**280.2**(8.1)	**174.5**(1.8)	**531.3**(18)	**201.6**(2.1)	**823.1**(46)	**207.6**(2.2)

8

Table S5: The setting is similar to the one used in Table S4: the eigenvalues follow an exponential with rate 1, the number of bases is 40 ($K = 20$) and the time points within any subject are dependently sampled; except that the signal-to-noise ratio is 0. Bold values indicate the top 2 performing methods.

	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	306.6(11)	191.7(2.1)	528.5(19)	203.5(2.2)	804.8(38)	**210.1**(2.2)
FACE	415.1(16)	202.9(2.2)	631.9(23)	206.6(2.2)	804.4(34)	**210.0**(2.2)
mFPCA	312.2(11)	196.1(2.1)	562.4(21)	206.0(2.2)	832.4(35)	212.1(2.2)
MICE	363.4(17)	137.4(1.6)	784.7(28)	253.7(2.4)	1385(56)	314.0(2.8)
CNP	765.3(39)	210.8(2.2)	707.0(33)	210.2(2.2)	953.4(47)	212.2(2.2)
GAIN	644.0(30)	307.8(3.8)	1563(57)	410.2(4.4)	1919(56)	431.4(2.6)
1DS	496.6(21)	179.5(2.1)	837.3(31)	254.1(3.0)	1120(43)	252.2(3.0)
GeLU-activated transformers with penalties						
GT1	**249.6**(9.4)	163.3(1.7)	506.7(17)	221.8(2.3)	799.0(45)	229.4(2.3)
GT1P	273.5(9.3)	185.9(2.1)	499.7(17)	201.3(2.1)	**780.4**(44)	210.6(2.2)
GT1S	**249.6**(9.4)	163.4(1.7)	506.7(17)	221.8(2.3)	797.3(45)	223.1(2.3)
GT2	251.8(9.9)	166.2(1.8)	511.1(17)	221.2(2.3)	800.7(48)	225.5(2.3)
GT2P	273.7(9.6)	186.1(2.1)	**503.9**(17)	**201.1**(2.2)	787.2(47)	210.5(2.5)
GT2S	251.8(9.9)	166.2(1.8)	511.1(17)	221.2(2.3)	800.7(48)	225.5(2.4)
GeLU-activated transformers with augmented modules						
ATT	251.5(9.1)	**160.5**(1.7)	508.5(17)	220.1(2.3)	853.6(42)	224.3(2.5)
SAND	**231.9**(8.5)	**159.1**(1.9)	**482.9**(17)	**196.2**(2.1)	**766.3**(40)	**210.1**(2.2)

Table S6: MSE (SE) and TV (SE) on the testing set. The eigenvalues follow an exponential distribution, the number of bases is 10 ($K = 5$) and the time points are sampled independently within any subject. Bold values indicate the top 2 performing methods.

	with measurement errors				without measurement errors			
	$n_i = 30$		$n_i = 8$ to 12		$n_i = 30$		$n_i = 8$ to 12	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	**37.85**(1.0)	**34.53**(0.5)	**254.2**(16)	**69.40**(1.6)	**2.889**(0.2)	**17.54**(0.4)	**173.9**(16)	**58.14**(1.6)
FACE	127.6(8.5)	64.87(1.6)	309.0(18)	77.78(1.7)	103.3(8.3)	61.52(1.6)	256.6(19)	73.09(1.8)
mFPCA	108.7(7.9)	61.66(1.6)	273.5(17)	72.94(1.7)	84.32(7.9)	57.86(1.7)	226.6(19)	67.21(1.8)
MICE	70.31(2.4)	153.5(1.1)	568.4(25)	223.7(2.8)	14.15(1.5)	30.71(0.6)	527.3(23)	202.5(2.7)
CNP	105.8(5.0)	65.8(0.88)	354.4(17)	85.58(1.7)	66.29(4.1)	56.93(0.8)	302.9(16)	82.01(1.6)
GAIN	86.37(2.6)	166.6(1.4)	906.4(37)	321.1(3.4)	43.13(2.3)	83.98(1.2)	883.5(37)	276.5(3.4)
1DS	115.0(7.6)	55.48(1.1)	524.7(22)	203.7(3.5)	25.61(4.5)	43.92(1.0)	452.4(21)	185.3(3.8)
Transformers								
GT	65.82(1.5)	132.9(0.92)	255.4(12)	110.3(1.6)	5.417(0.37)	30.75(0.6)	**173.9**(11)	86.54(1.5)
SAND	**36.32**(1.2)	**34.41**(0.46)	**218.4**(12)	**72.85**(1.3)	**2.685**(0.2)	**9.678**(0.25)	**149.7**(10)	**54.03**(1.3)

9

Table S7: MSE (SE) and TV (SE) on the testing set. The eigenvalues follow a t-distribution, the number of bases is 40 ($K = 20$) and the time points are sampled independently within any subject. Bold values indicate the top 2 performing methods.

	with measurement errors				without measurement errors			
	$n_i = 30$		$n_i = 8$ to 12		$n_i = 30$		$n_i = 8$ to 12	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	317.4(7.2)	**244.1**(2.3)	**744.9**(20)	**265.2**(2.4)	254.2(6.8)	239.1(2.3)	671.3(20)	**263.7**(2.4)
FACE	472.9(11)	260.8(2.3)	796.3(22)	267.6(2.4)	434.3(11)	259.9(2.3)	729.2(22)	266.5(2.4)
mFPCA	440.3(11)	258.5(2.3)	762.5(21)	265.6(2.4)	401.0(10)	257.4(2.3)	695.3(21)	264.2(2.4)
MICE	304.9(5.8)	300.6(1.9)	1258(40)	419.6(3.4)	188.9(4.9)	**190.6**(1.7)	1200(39)	395.7(3.4)
CNP	519.1(12)	262.7(2.3)	984.1(30)	273.5(2.3)	410.4(12)	257.7(2.3)	920.1(27)	273.4(2.3)
GAIN	376.1(7.7)	377.6(2.8)	2019(68)	511.5(4.6)	299.7(7.6)	303.4(2.9)	1893(63)	488.9(4.5)
1DS	437.8(8.2)	351.6(2.7)	1229(34)	398.3(3.9)	265.2(6.9)	252.1(2.4)	1093(35)	374.8(4.0)
				Transformers				
GT	**297.5**(6.0)	284.5(1.9)	776.2(21)	289.0(2.3)	**171.6**(4.6)	201.7(1.8)	**666.1**(19)	271.1(2.3)
SAND	**264.0**(5.4)	**215.7**(1.9)	**731.1**(20)	**259.5**(2.2)	**174.5**(4.5)	**197.6**(2.1)	**637.2**(18)	**253.7**(2.3)

Table S8: MSE (SE) and TV (SE) on the testing set. The eigenvalues follow a t-distribution, the number of bases is 10 ($K = 5$) and the time points are sampled independently within any subject. Bold values indicate the top 2 performing methods.

	with measurement errors				without measurement errors			
	$n_i = 30$		$n_i = 8$ to 12		$n_i = 30$		$n_i = 8$ to 12	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	**64.88**(1.6)	**45.28**(0.6)	**397.4**(14)	**93.13**(1.6)	3.352(0.2)	**21.43**(0.3)	**278.3**(13)	**82.98**(1.7)
FACE	206.6(7.3)	87.41(1.6)	503.2(22)	103.0(1.7)	168.9(7.2)	83.67(1.7)	414.1(20)	98.21(1.8)
mFPCA	175.6(6.5)	83.99(1.6)	435.3(15)	98.08(1.7)	136.4(6.4)	79.65(1.7)	366.4(17)	92.31(1.9)
MICE	131.9(13)	203.8(2.6)	1074(68)	306.1(3.6)	40.22(13)	43.93(2.5)	1015(66)	278.2(3.7)
CNP	272.0(62)	92.32(1.2)	767.0(76)	117.7(1.7)	242.3(100)	81.27(1.3)	684.0(51)	115.5(1.8)
GAIN	171.1(15)	225.0(3.8)	2166(140)	417.2(5.7)	97.85(13)	121.5(3.6)	1663(110)	361.6(4.7)
1DS	184.9(6.9)	73.09(1.2)	981.2(71)	270.0(4.4)	42.20(3.1)	59.30(1.1)	854.7(70)	239.8(4.5)
				Transformers				
GT	126.6(4.1)	176.3(1.1)	480.7(22)	146.3(1.9)	13.98(4.2)	40.30(0.8)	336.7(19)	121.0(1.9)
SAND	**80.58**(2.8)	**53.6**(0.65)	**424.2**(18)	**96.35**(1.4)	**9.181**(4.0)	**13.00**(0.5)	**295.0**(17)	**74.47**(1.5)

Table S9: MSE (SE) and TV (SE) on the testing set. The eigenvalues follow a standard Gaussian, the number of bases is 40 ($K = 20$) and the time points are sampled independently within any subject. Bold values indicate the top 2 performing methods.

	with measurement errors				without measurement errors			
	$n_i = 30$		$n_i = 8$ to 12		$n_i = 30$		$n_i = 8$ to 12	
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	183.4(2.6)	**186.9**(1.1)	**451.0**(10)	**203.3**(1.1)	145.3(2.3)	183.2(1.1)	**397.8**(10)	**201.7**(1.1)
FACE	276.6(4.3)	199.9(1.1)	496.9(11)	205.9(1.1)	255.6(4.2)	199.3(1.1)	451.3(11)	205.0(1.1)
mFPCA	258.9(4.2)	198.2(1.1)	466.9(11)	204.3(1.1)	236.9(4.2)	197.7(1.1)	422.8(11)	203.1(1.2)
MICE	189.2(2.7)	231.0(0.4)	816.5(20)	327.5(0.6)	112.1(2.6)	145.5(0.3)	773.7(19)	308.3(0.6)
CNP	308.1(21)	201.0(1.2)	636.4(24)	210.6(1.2)	247.9(7.1)	198.2(1.2)	570.5(17)	209.5(1.1)
GAIN	241.4(4.2)	288.4(1.5)	1535(45)	392.8(3.9)	159.1(2.9)	221.3(1.4)	1525(45)	383.5(3.9)
1DS	255.3(4.1)	271.3(1.7)	797.6(19)	313.4(2.9)	157.7(3.4)	196.7(1.4)	705.4(18)	291.9(2.7)
				Transformers				
GT	**174.3**(2.7)	222.6(1.1)	485.4(12)	220.4(1.2)	**99.32**(2.1)	**150.8**(1.0)	418.6(11)	209.9(1.2)
SAND	**154.9**(2.6)	**167.1**(1.1)	**456.5**(11)	**199.6**(1.1)	**103.4**(2.1)	**153.3**(1.1)	**397.4**(10)	**194.1**(1.2)

10

Table S10: MSE (SE) and TV (SE) on the testing set. The eigenvalues follow a standard Gaussian, the number of bases is 10 ($K = 5$) and the time points are sampled independently within any subject. Bold values indicate the top 2 performing methods.

| | with measurement errors | | | | without measurement errors | | | |
| | $n_i = 30$ | | $n_i = 8$ to 12 | | $n_i = 30$ | | $n_i = 8$ to 12 | |
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	**39.01**(0.9)	**34.55**(0.4)	**227.6**(7.2)	**70.66**(1.0)	**1.351**(0.1)	**15.56**(0.2)	**142.6**(6.0)	**58.14**(1.0)
FACE	124.6(3.6)	68.74(1.1)	314.7(9.6)	81.39(1.1)	101.8(3.5)	66.12(1.1)	269.3(9.5)	77.74(1.2)
mFPCA	106.5(3.3)	65.76(1.0)	262.1(8.3)	77.37(1.1)	82.07(3.2)	62.51(1.1)	222.7(9.1)	73.06(1.3)
MICE	70.44(1.6)	153.2(0.9)	618.3(17)	234.3(2.0)	13.26(1.0)	29.73(0.5)	576.0(16)	212.8(1.9)
CNP	102.4(5.0)	68.37(0.6)	398.7(12)	91.43(1.0)	69.82(4.5)	62.35(0.7)	346.1(11)	90.15(1.1)
GAIN	80.49(1.9)	160.9(1.1)	1355(42)	309.0(4.1)	31.63(1.4)	68.36(1.0)	1019(26)	293.9(2.9)
1DS	123.4(10)	57.93(0.8)	549.4(16)	213.0(2.8)	25.58(2.2)	46.79(0.7)	470.3(16)	190.0(2.7)
	Transformers							
GT	75.86(1.2)	140.9(0.9)	283.3(8.6)	111.1(1.0)	6.418(0.4)	34.13(0.4)	191.8(7.4)	91.76(1.0)
SAND	**50.08**(1.1)	**42.87**(0.5)	**253.8**(8.3)	**77.01**(0.9)	**3.413**(0.3)	**11.71**(0.2)	**168.4**(7.2)	**58.12**(0.9)

Table S11: MSE(SE) and TV(SE) on the UK electricity dataset when t_{ij} are sampled dependently. Bold values indicate the top 2 performing methods.

| | UK electricity | | | | | |
| | $n_i = 30$ | | $n_i = 8$ to 12 | | $n_i = 3, 4, 5$ | |
	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)	MSE(SE)	TV(SE)
PACE	18.24(2.9)	**19.36**(1.1)	30.17(4.0)	**21.15**(1.2)	46.34(5.8)	**22.25**(1.2)
FACE	27.52(4.2)	21.93(1.2)	38.61(5.4)	22.78(1.2)	48.19(6.2)	23.47(1.2)
mFPCA	20.67(3.1)	22.70(1.2)	32.45(4.3)	24.39(1.2)	**46.26**(6.1)	23.63(1.2)
MICE	24.49(3.2)	59.05(2.6)	44.39(5.9)	57.18(2.3)	74.86(8.5)	68.06(1.7)
CNP	26.06(3.8)	21.43(1.2)	36.73(4.7)	22.17(1.2)	65.11(12)	**22.80**(1.2)
GAIN	35.70(4.9)	84.94(4.1)	56.27(7.8)	82.33(3.3)	124.9(16)	52.43(2.7)
1DS	58.51(7.7)	49.53(2.6)	83.27(11)	38.81(2.2)	108.4(13)	36.54(2.2)
GT	**16.17**(2.8)	22.04(1.2)	**25.77**(3.5)	25.41(1.4)	49.69(7.0)	40.92(2.7)
SAND	**15.69**(2.8)	**17.85**(1.1)	**24.09**(3.4)	**20.54**(1.2)	**45.50**(6.4)	29.40(1.9)

11

References

[1] James, G. M., T. J. Hastie, and C. A. Sugar (2000). Principal component models for sparse functional data. *Biometrika 87*(3), 587–602.

[2] Peng, J. and D. Paul (2009). A geometric approach to maximum likelihood estimation of the functional principal components from sparse longitudinal data. *Journal of Computational and Graphical Statistics 18*(4), 995–1015.

[3] Rice, J. A. and C. O. Wu (2001). Nonparametric mixed effects models for unequally sampled noisy curves. *Biometrics 57*(1), 253–259.

12