SAND

Smooth Imputation of Sparse And Noisy Functional Data With Transformer Networks

Ju-Sheng Hong[1], Junwen Yao[1], Jonas Mueller[2], Jane-Ling Wang[1]

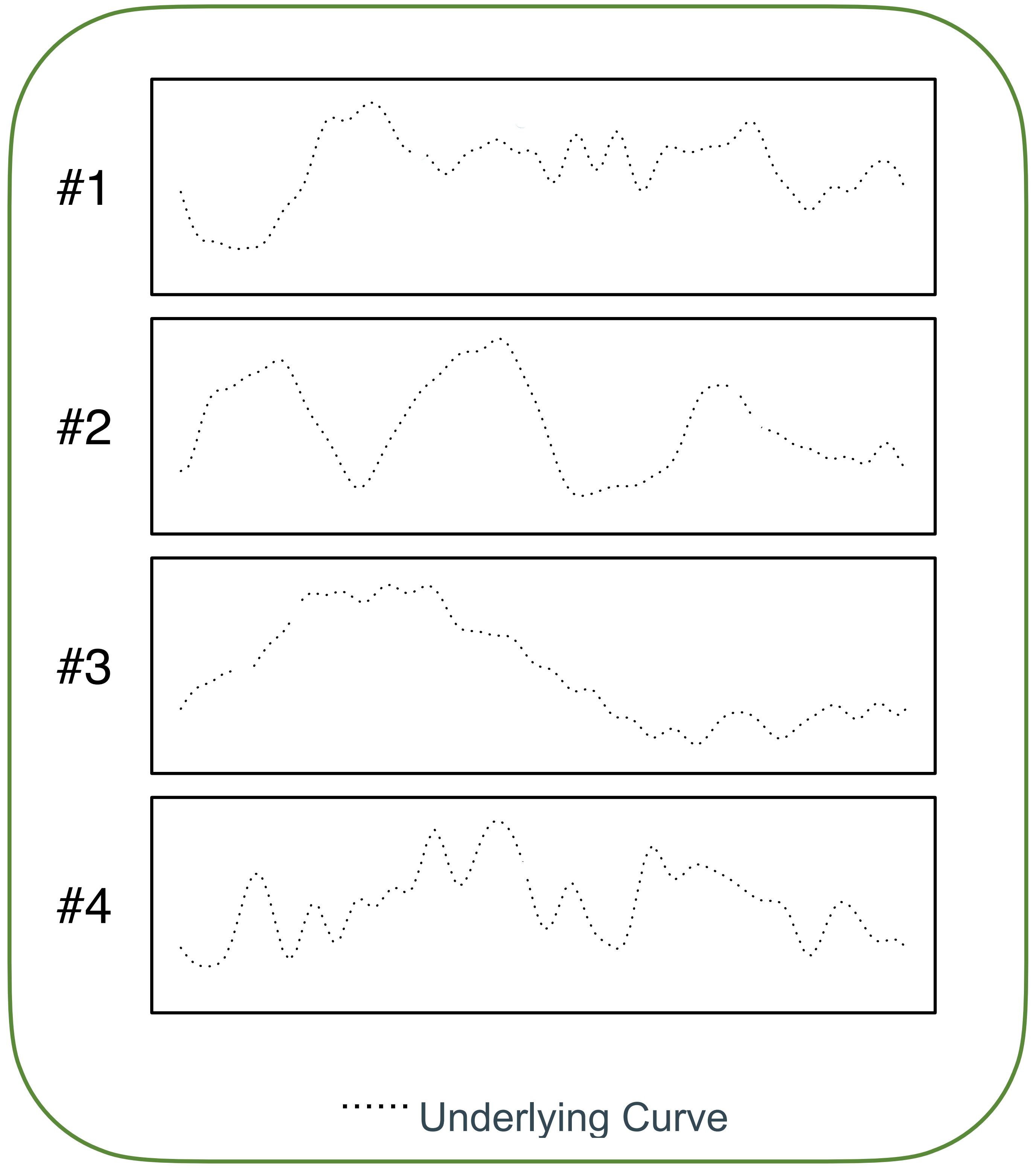
[1]University of California Davis
[2]Cleanlab

Sparse and Noisy Functional Data

- Functional data are random functions $X_i(t), t \in [0,1]$.

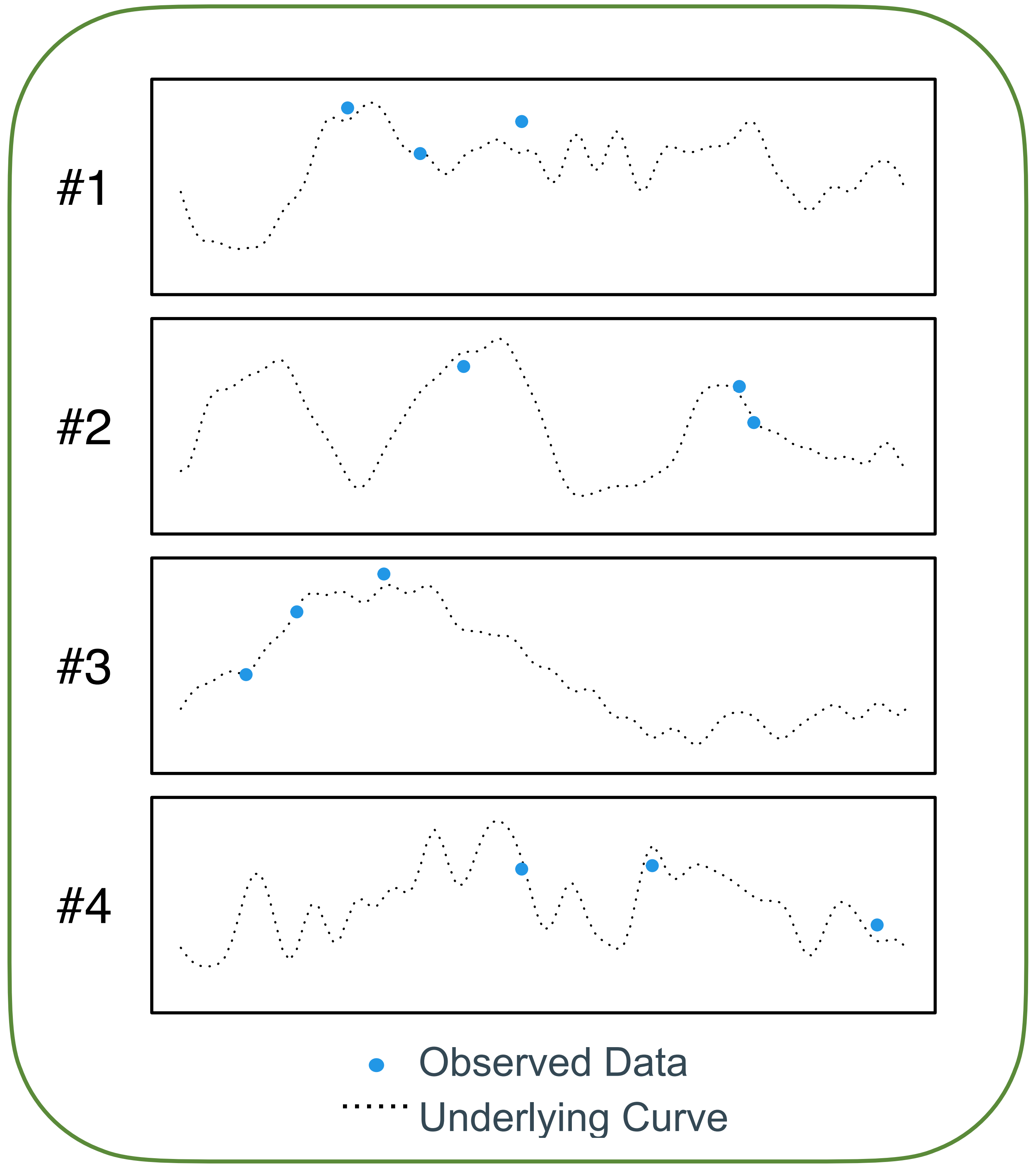
Sparse and Noisy Functional Data

- Functional data are random functions $X_i(t), t \in [0,1]$.



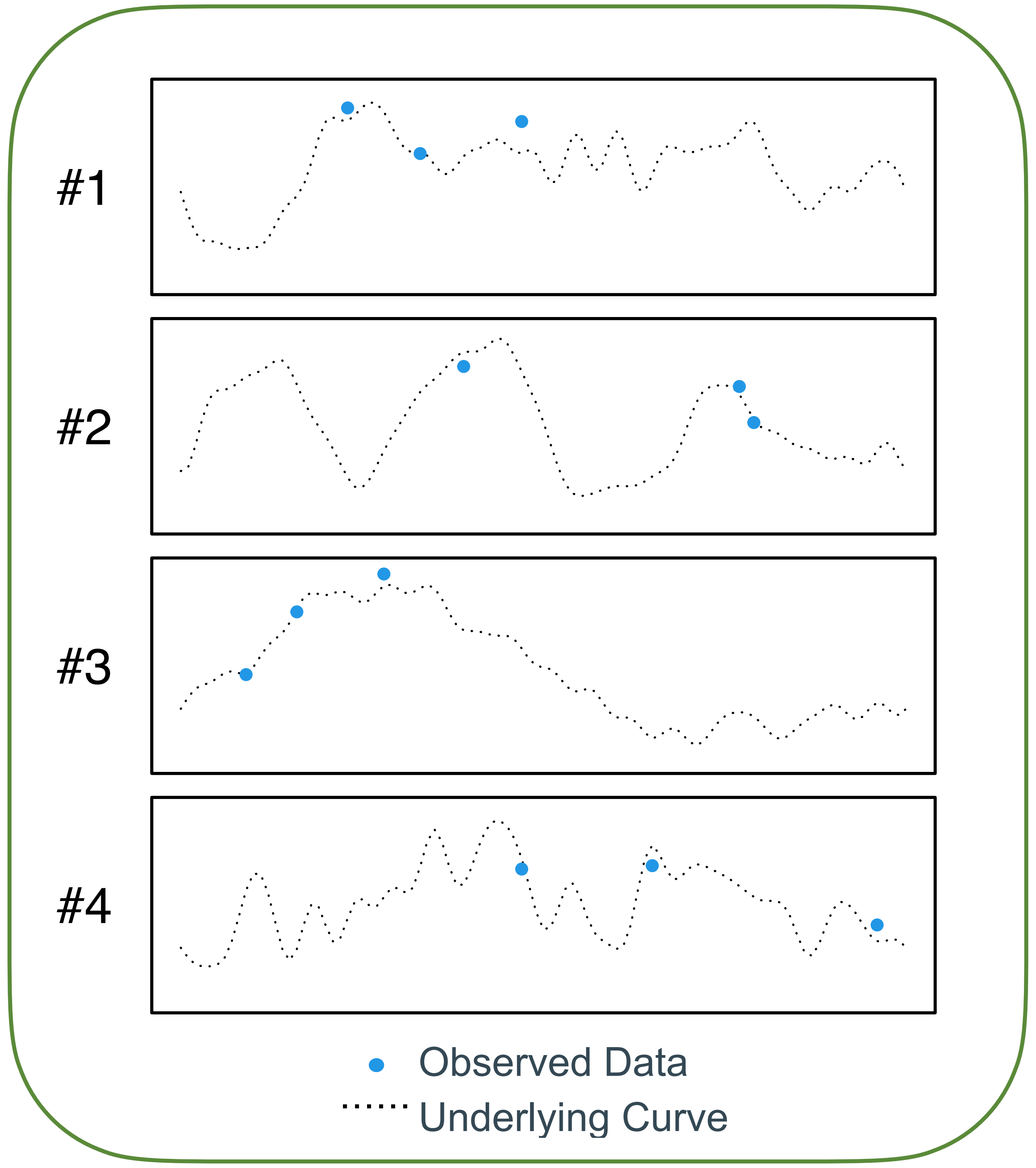
Sparse and Noisy Functional Data

- Functional data are random functions $X_i(t), t \in [0,1]$.
- $X_i(\cdot)$ is observed at time $t_{i1}, \ldots, t_{in_i}$



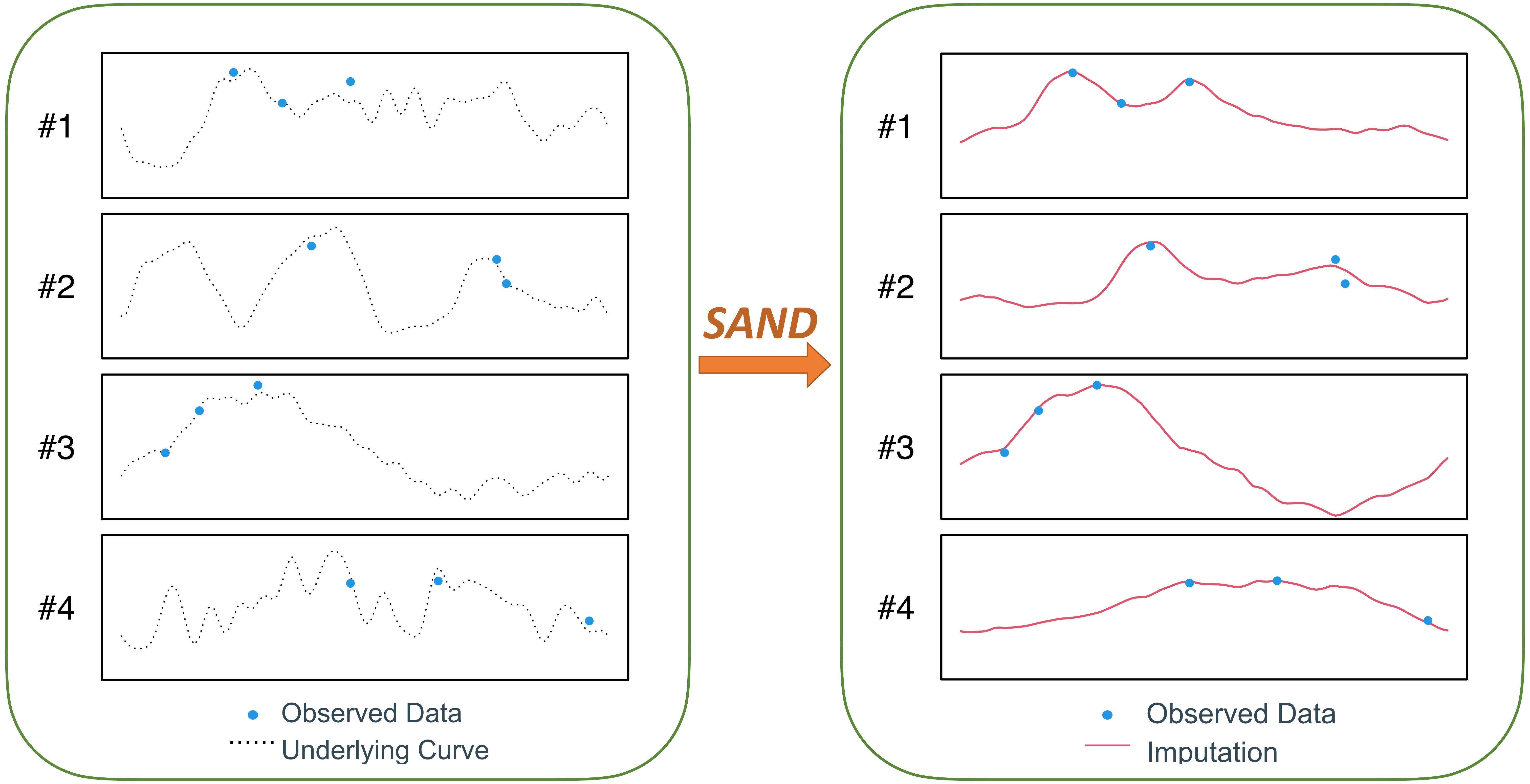
Sparse and Noisy Functional Data

- Functional data are random functions $X_i(t), t \in [0,1]$.
- $X_i(\cdot)$ is observed at time $t_{i1}, \ldots, t_{in_i}$
- Observations: $Y_{ij} = X(t_{ij}) + \varepsilon_{ij}$.

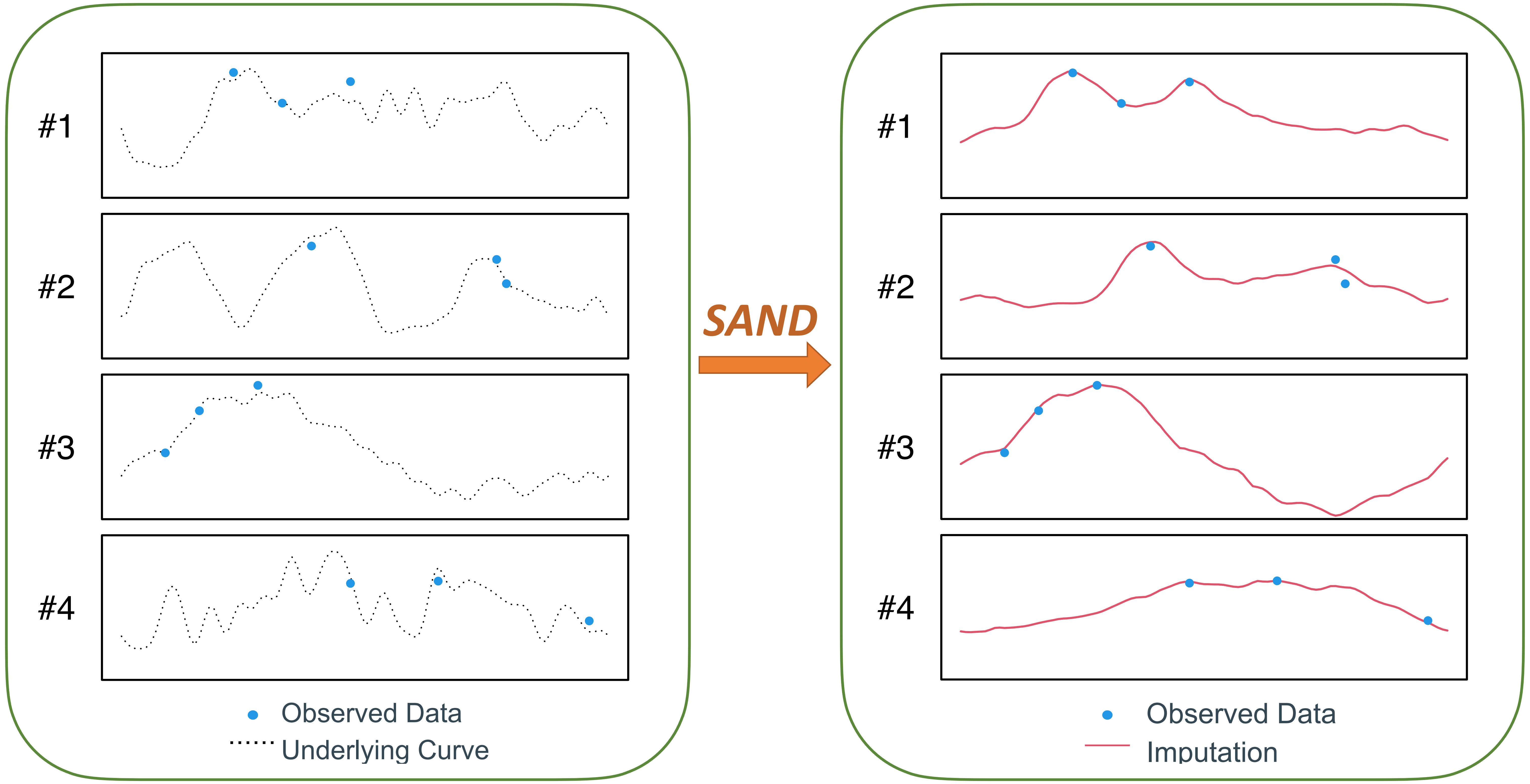


Sparse and Noisy Functional Data

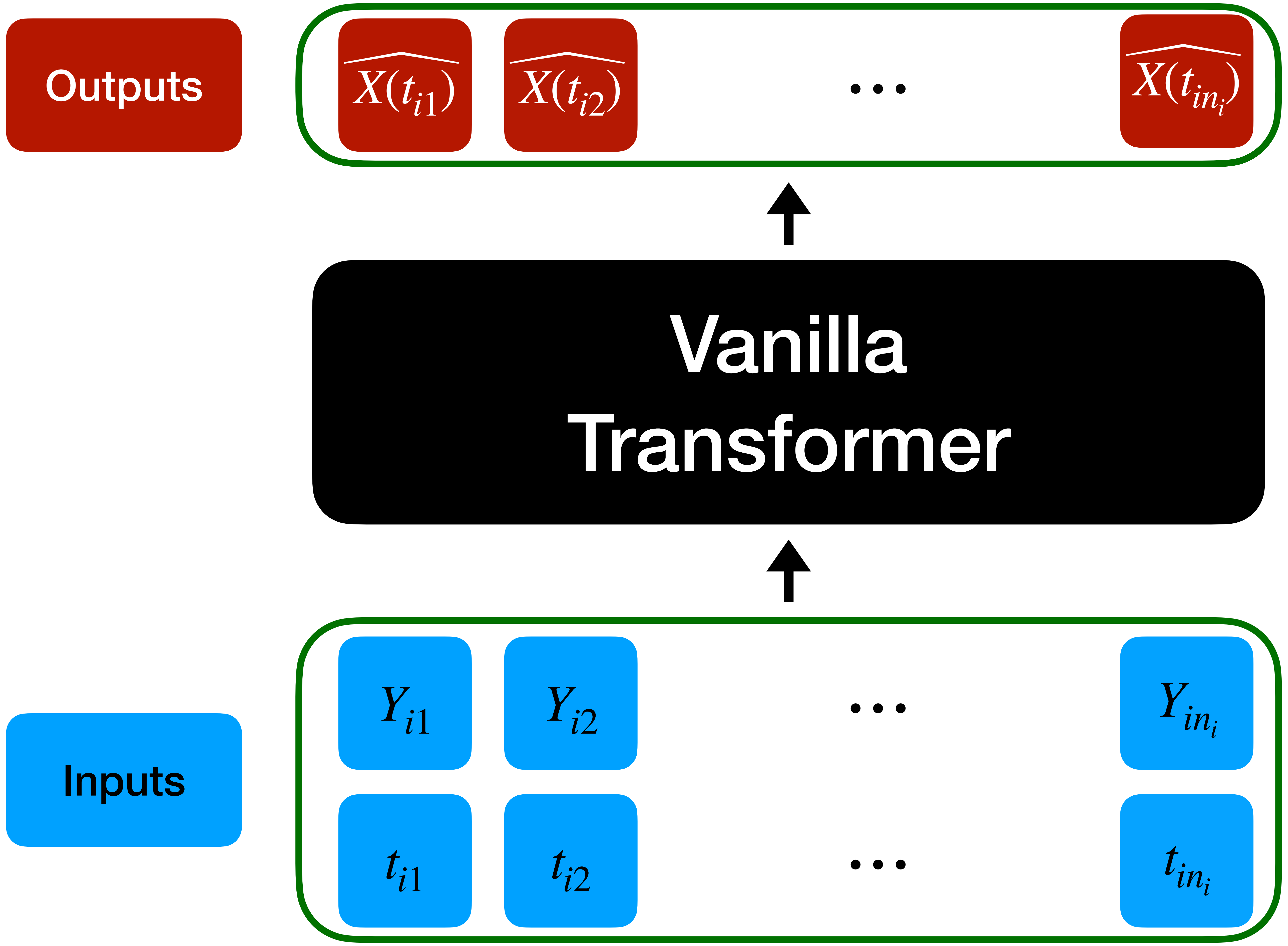
- $X_i(\cdot)$ is observed at time $t_{i1}, \ldots, t_{in_i}$
- Observations: $Y_{ij} = X(t_{ij}) + \varepsilon_{ij}$.



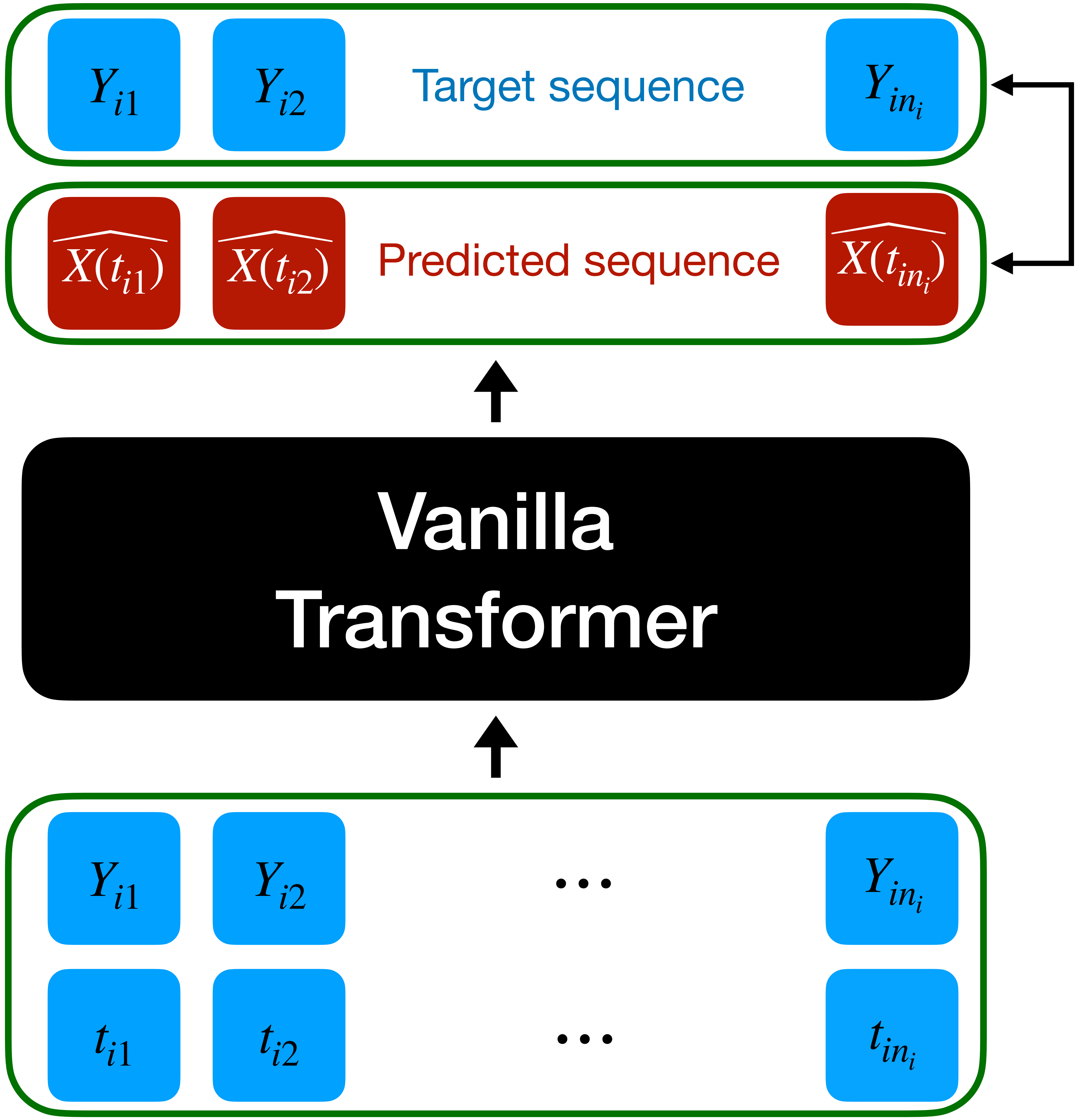
Goal: Recovering Underlying Curves



Limitations of Vanilla Transformers for Imputation



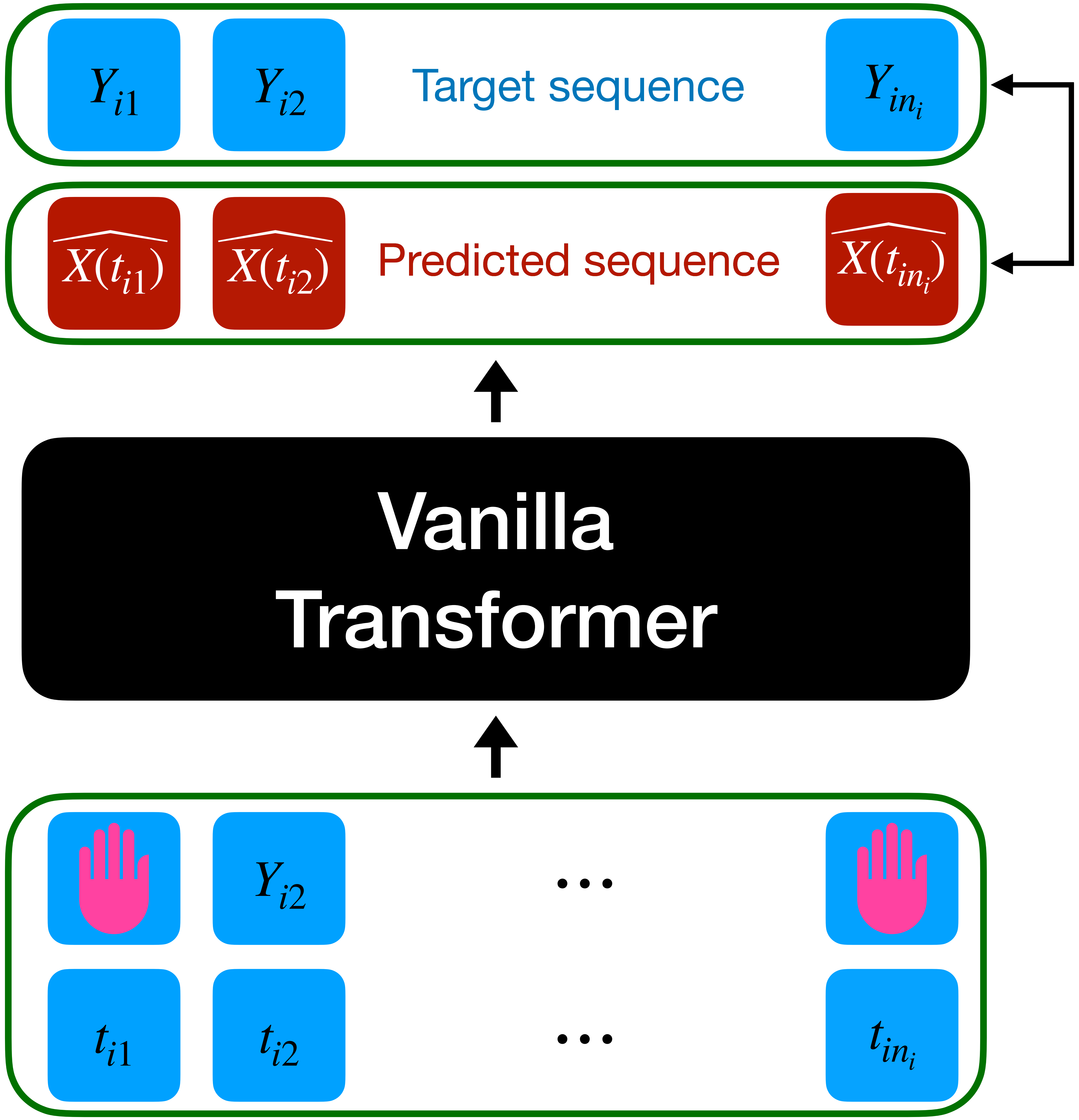
Limitations of Vanilla Transformers for Imputation



Minimizing the ℓ_2 distance between them

$$\text{Loss} = \sum_i \sum_{j=1}^{n_i} [\widehat{X}(t_{ij}) - Y_{ij}]^2$$

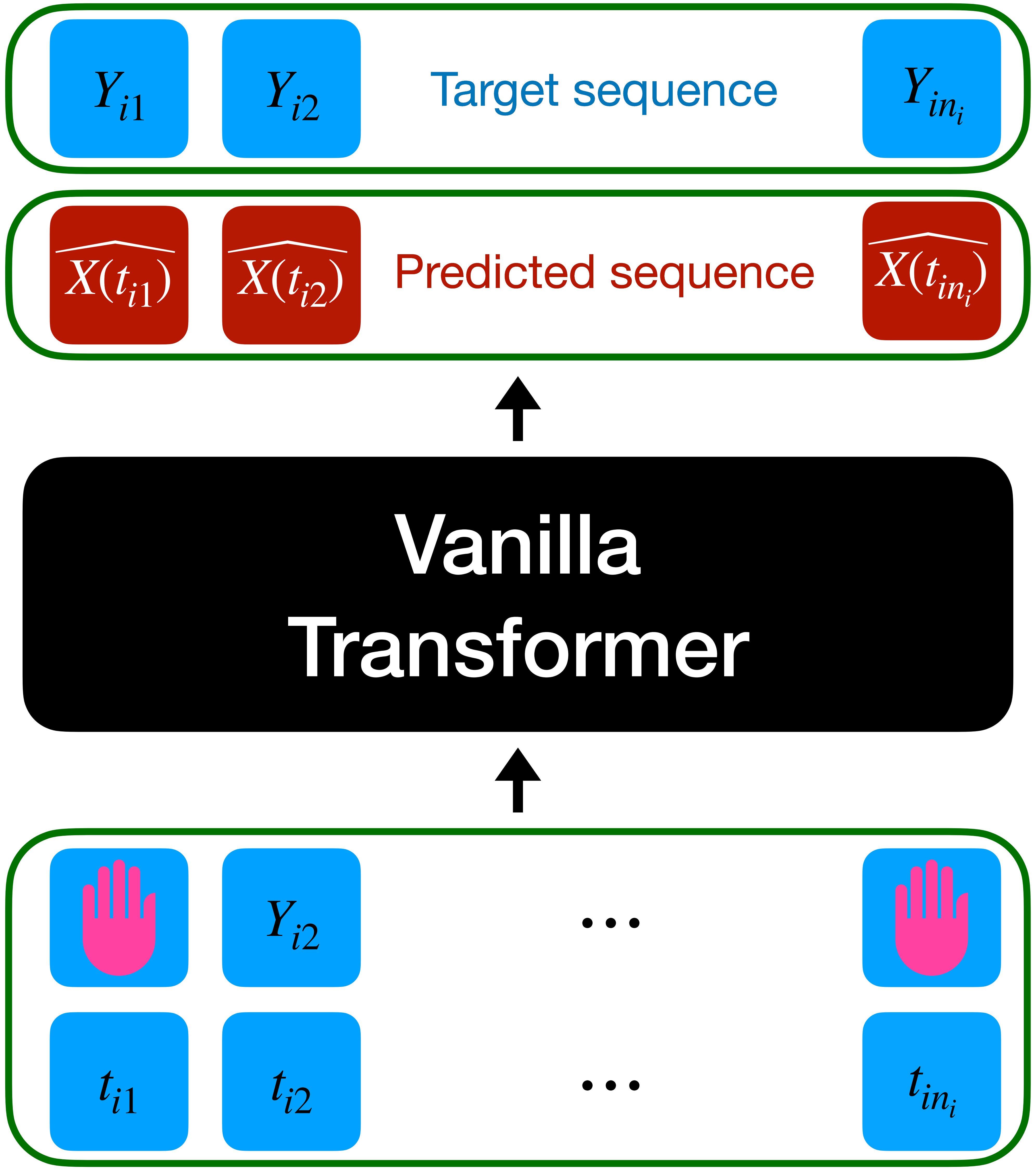
Limitations of Vanilla Transformers for Imputation

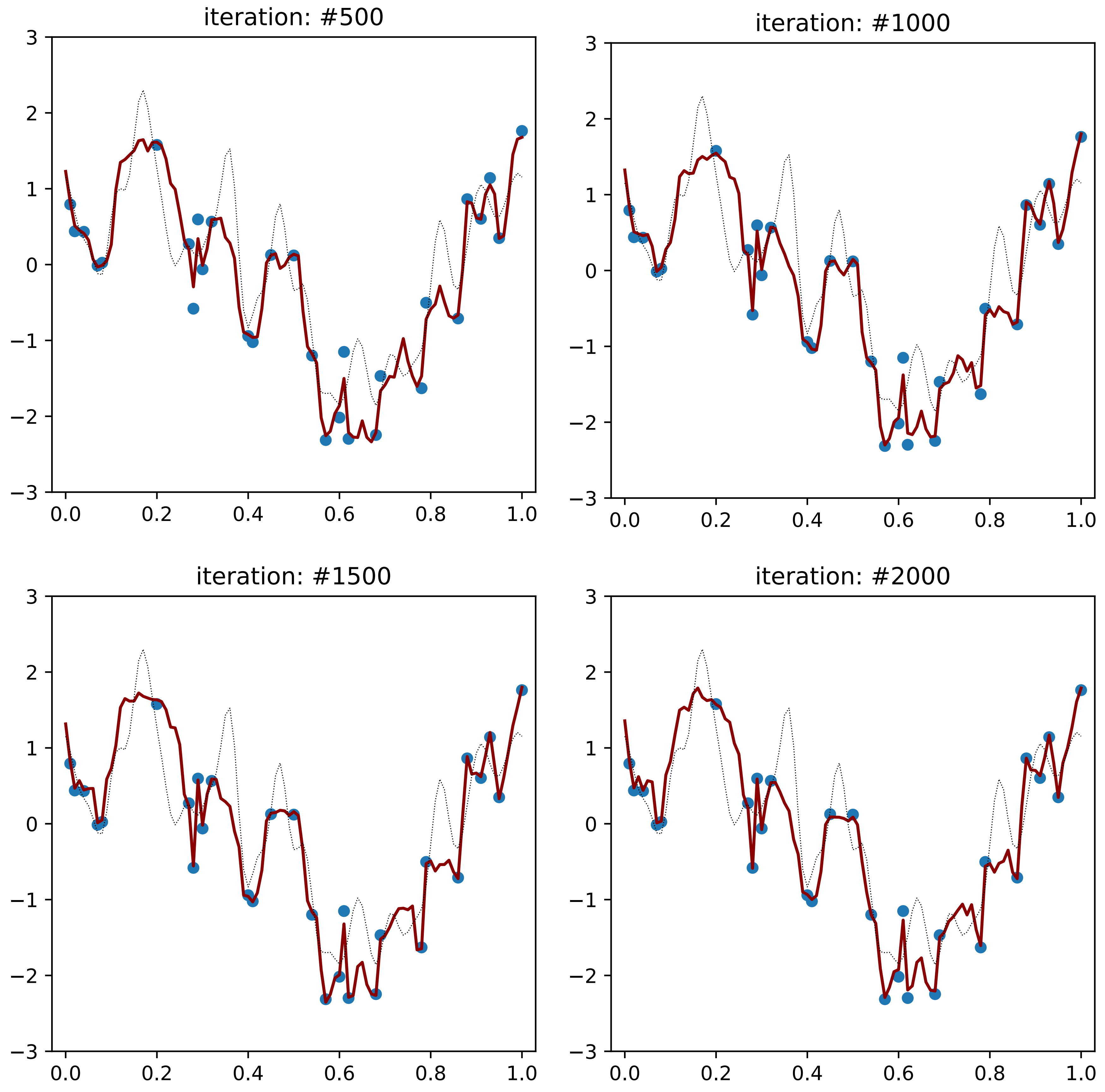


Minimizing the ℓ_2 distance between them

$$\text{Loss} = \sum_i \sum_{j=1}^{n_i} [\widehat{X}(t_{ij}) - Y_{ij}]^2$$

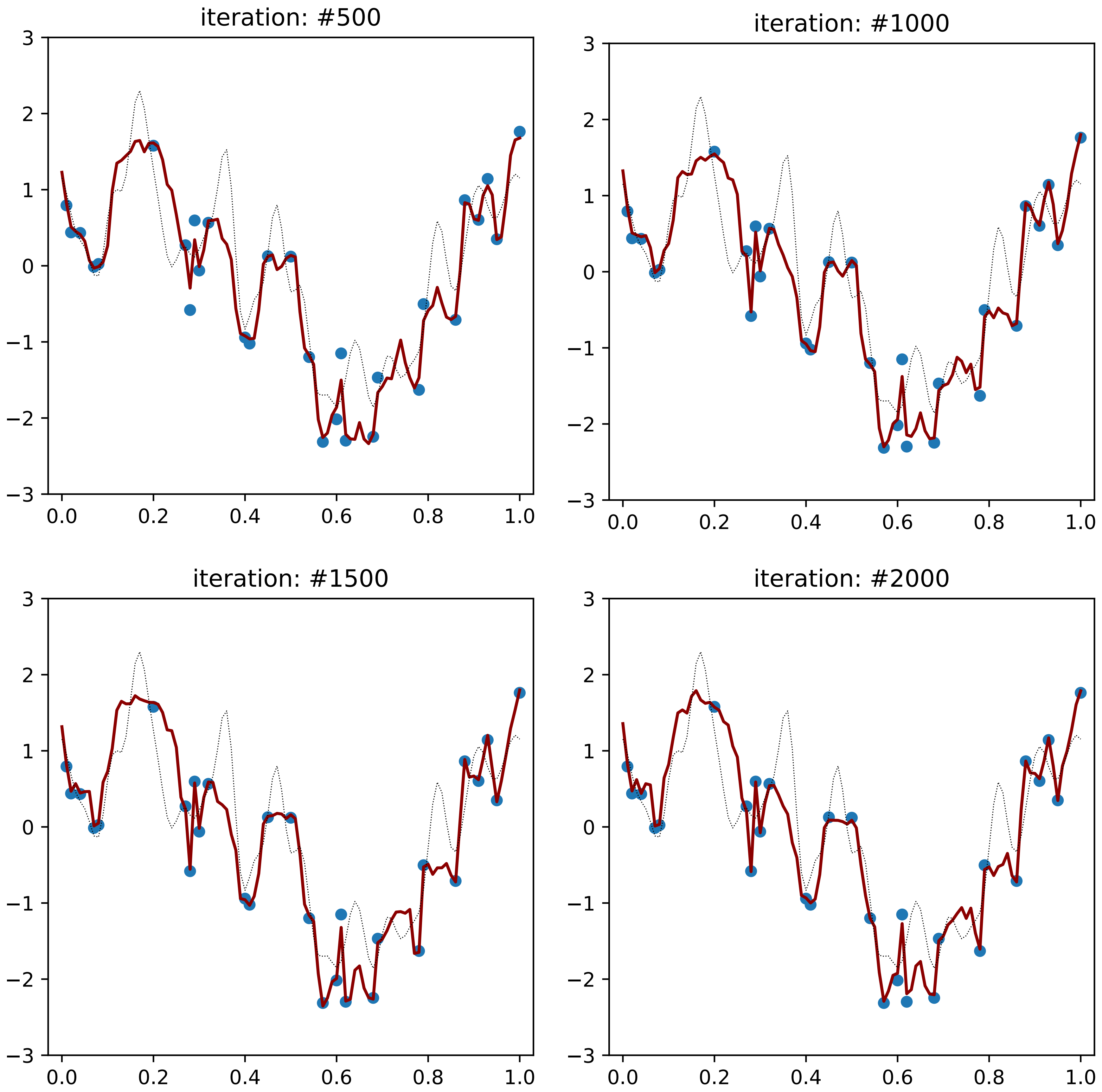
Limitations of Vanilla Transformers for Imputation





Vanilla Transformer:
Imputation Progress On **Testing Data** Over Iterations

Why vanilla transformers struggle with noisy data?

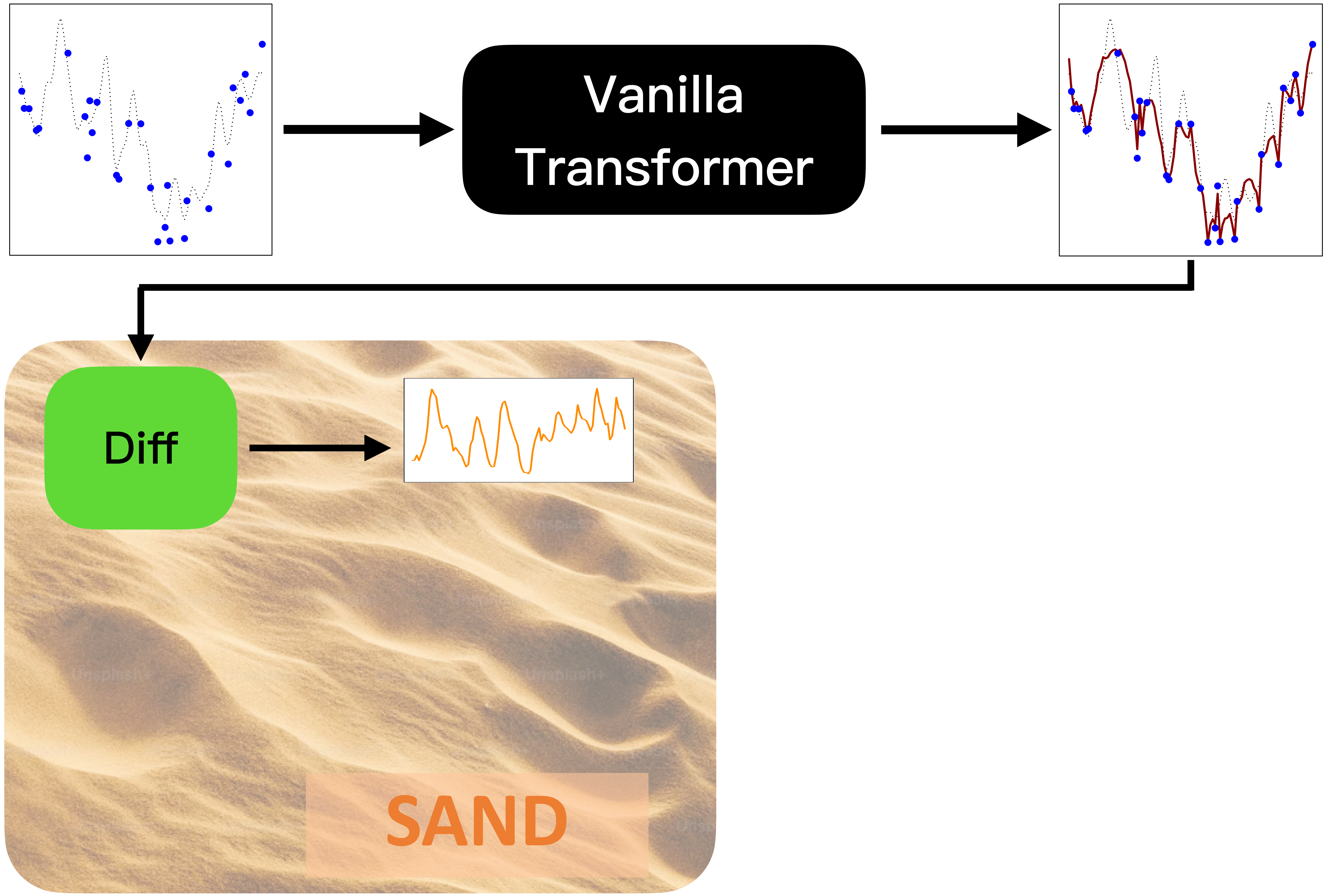


Vanilla Transformer:
Imputation Progress On **Testing Data** Over Iterations

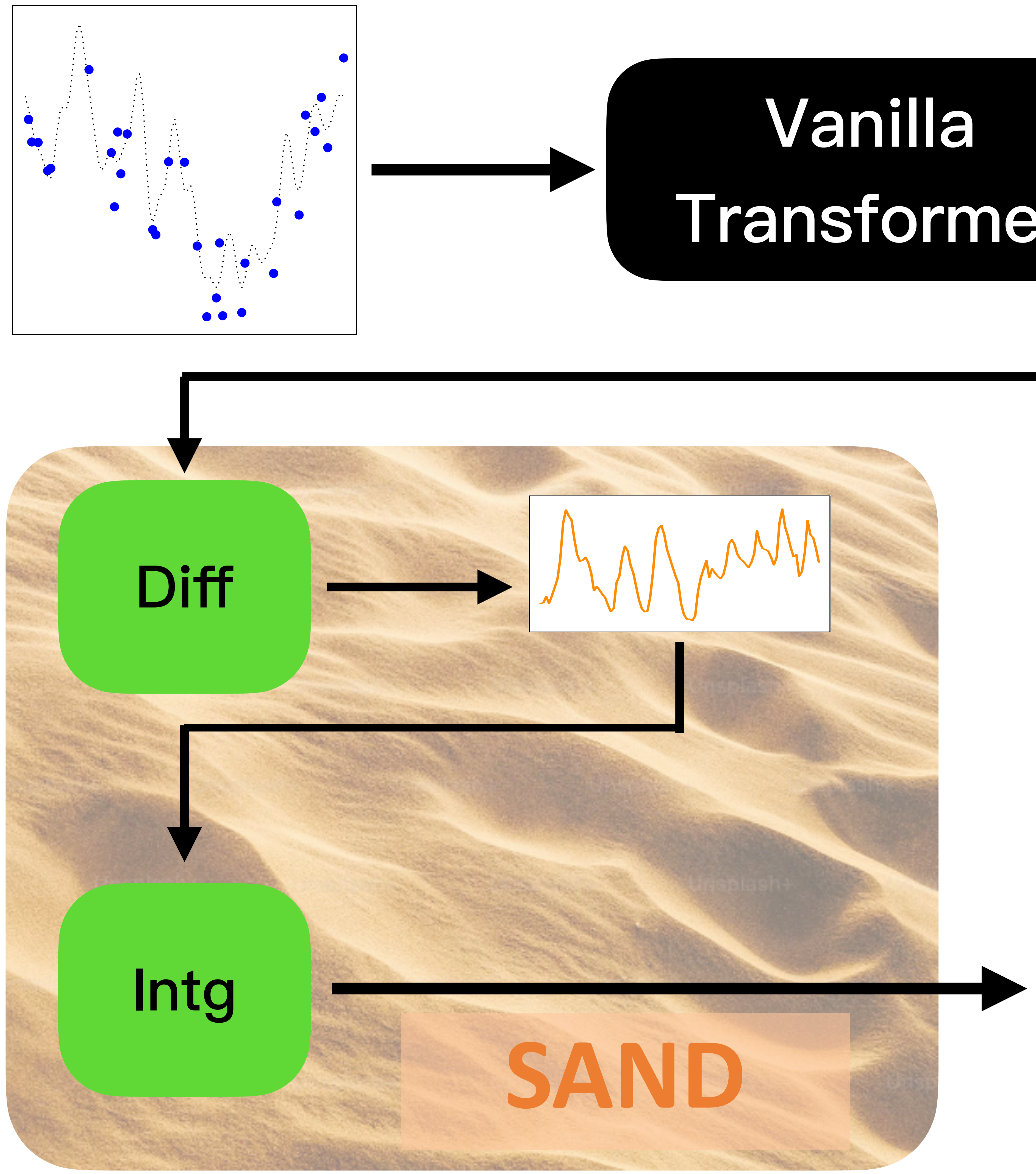
- Transformers are universal approximators [4].
- Training data Y_{ij} are noisy.
- Imputed data mimics noise patterns

SAND — Self AtteNtion on Derivative

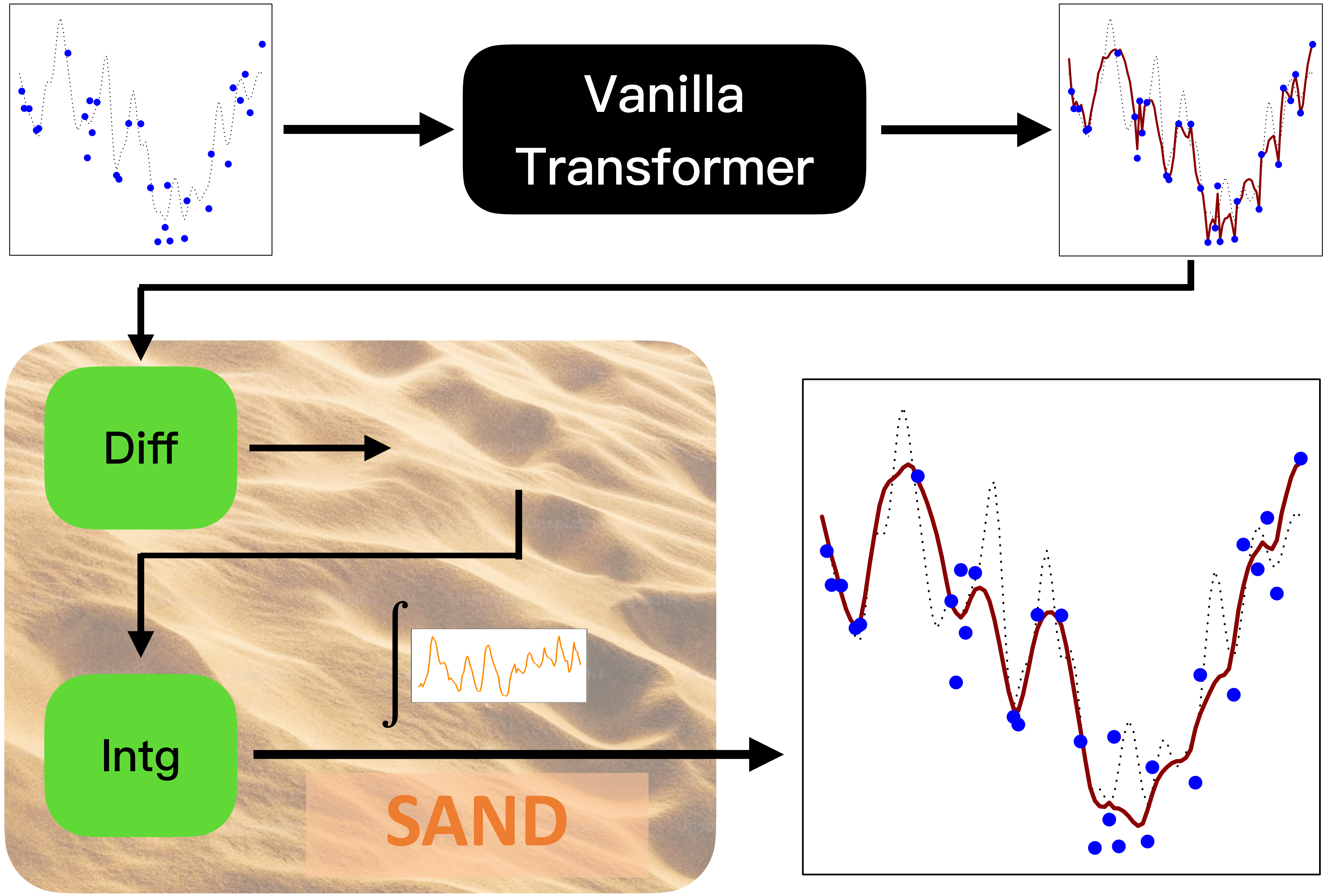
SAND — Self AtteNtion on Derivative



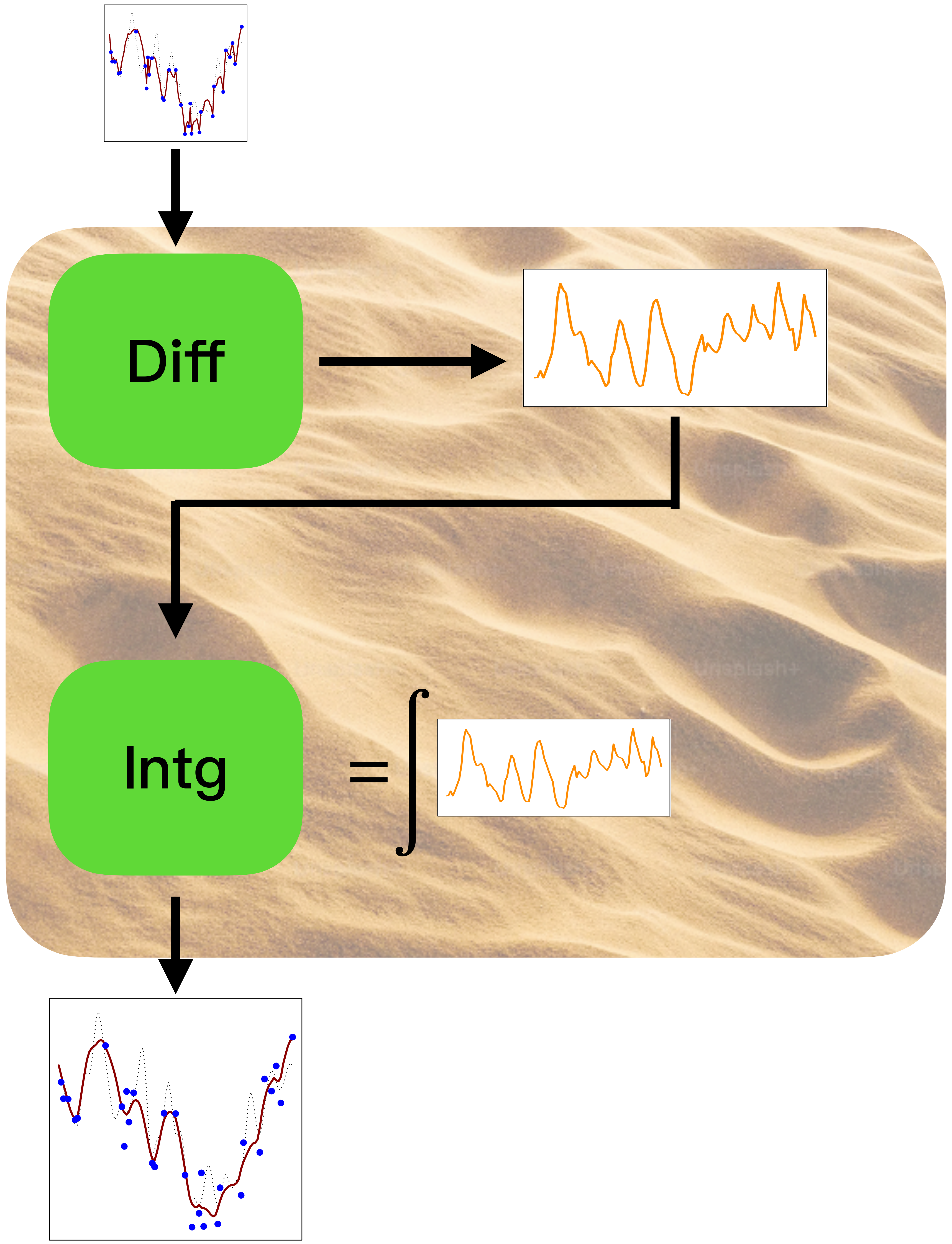
SAND — Self AtteNtion on Derivative



SAND — Self AtteNtion on Derivative

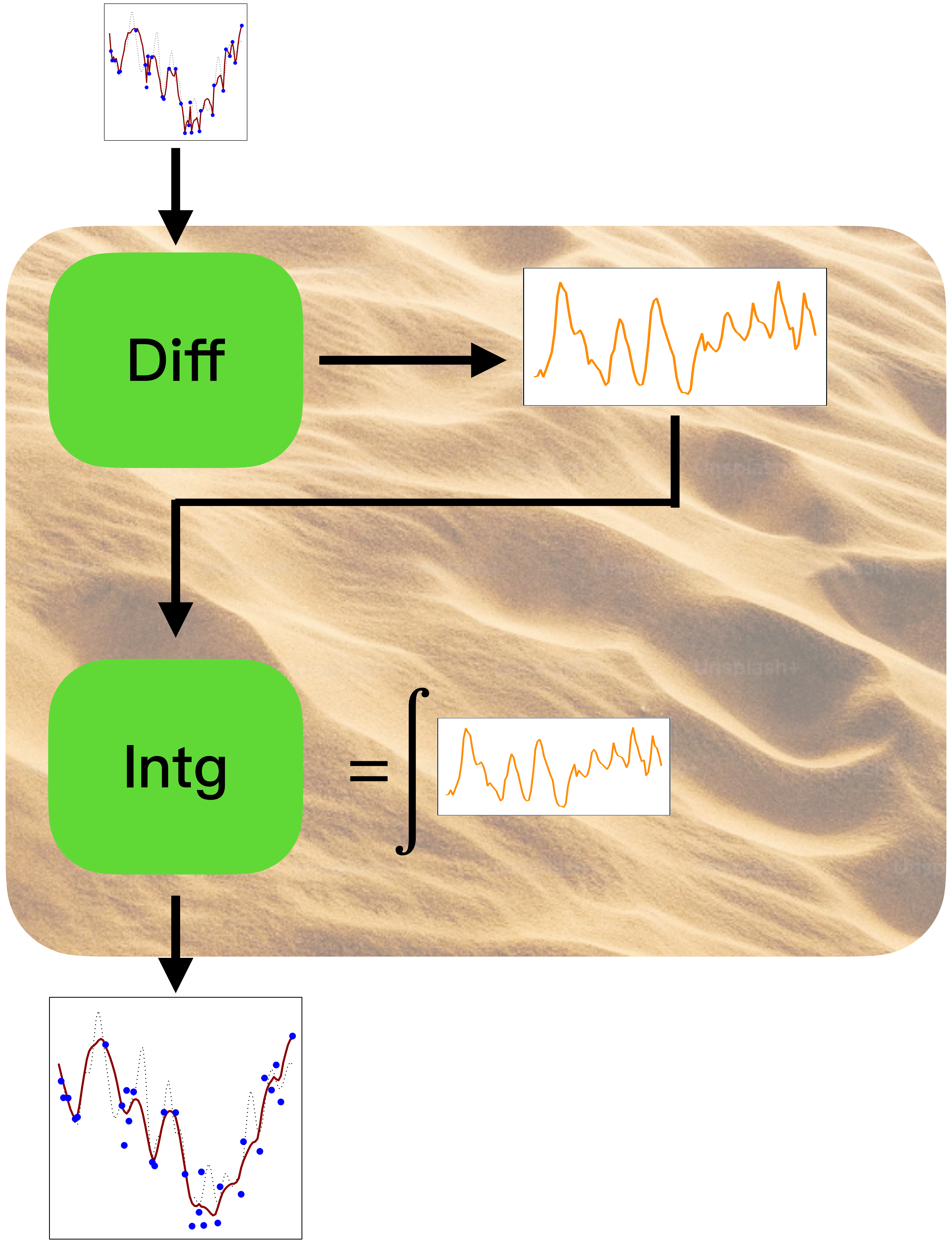


SAND — Self AtteNtion on Derivative



Input: $\widetilde{T}$, an coarse imputation from a vanilla transformer

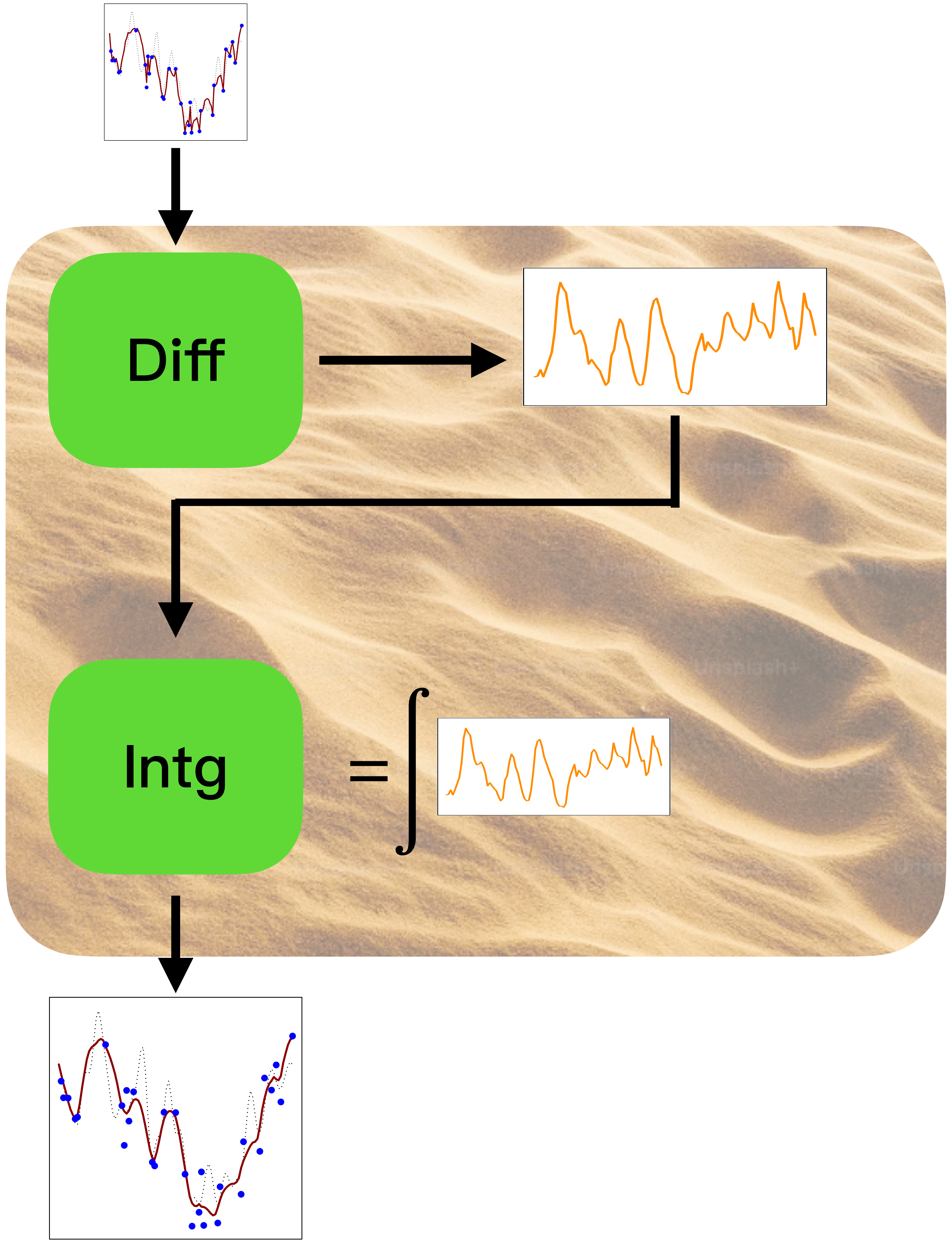
SAND — Self AtteNtion on Derivative



Input: $\widetilde{T}$, an coarse imputation from a vanilla transformer

$$\mathrm{Diff}(\widetilde{T}) = \sum_{h=1}^{H} W_O^{(h)} \left(W_V^{(h)} \widetilde{T} \right) \left[\left(W_K^{(h)} \widetilde{T} \right)^{\mathsf{T}} \left(W_Q^{(h)} \widetilde{T} \right) \Big/ \sqrt{h_d} \right]$$

SAND — Self AtteNtion on Derivative

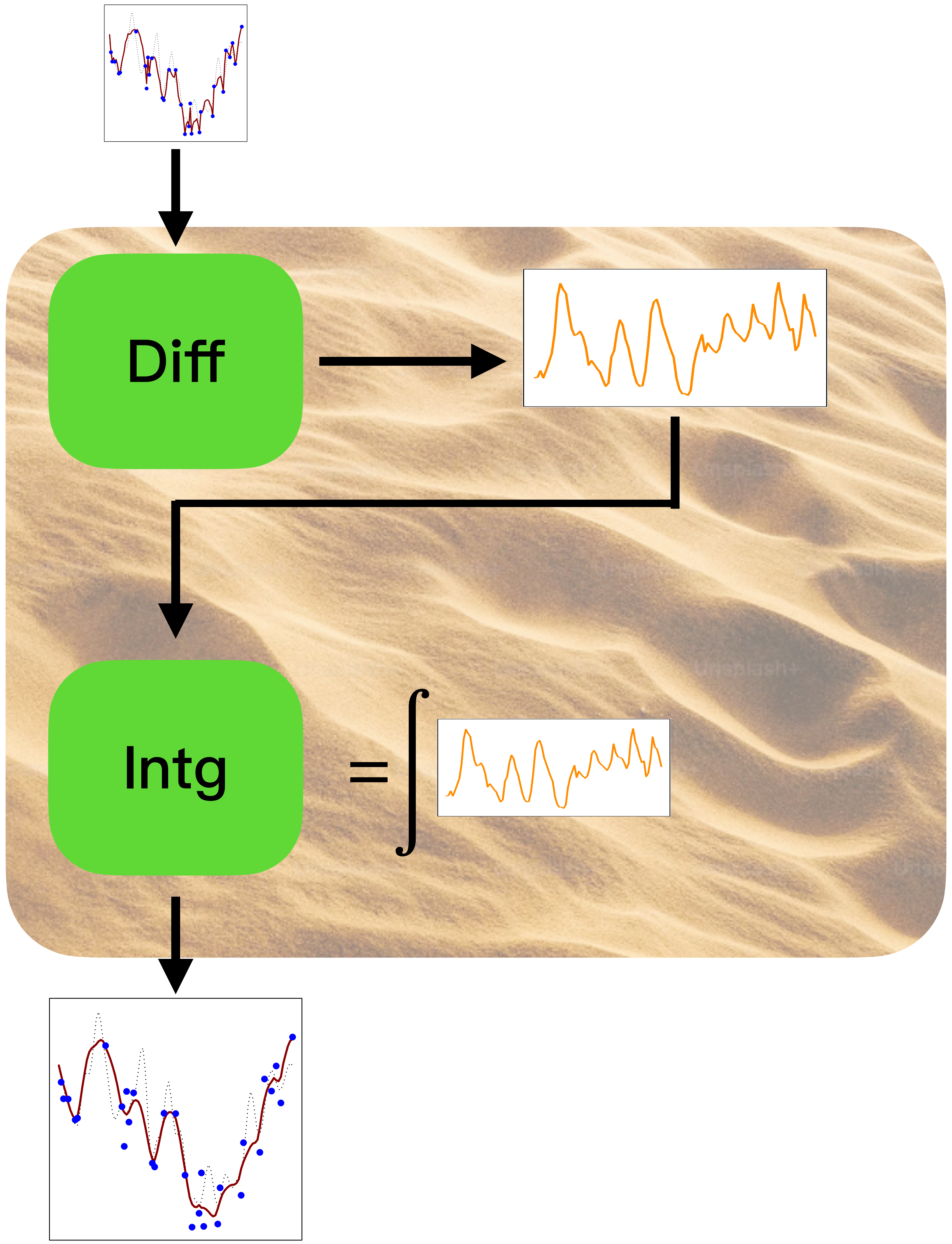


Input: $\widetilde{T}$, an coarse imputation from a vanilla transformer

$$\mathrm{Diff}(\widetilde{T}) = \sum_{h=1}^{H} W_O^{(h)} \left(W_V^{(h)} \widetilde{T} \right) \left[\left(W_K^{(h)} \widetilde{T} \right)^{\top} \left(W_Q^{(h)} \widetilde{T} \right) \Big/ \sqrt{h_d} \right]$$

Intg is the cumulative summation operator.

SAND — Self AtteNtion on Derivative



Input: $\widetilde{T}$, an coarse imputation from a vanilla transformer

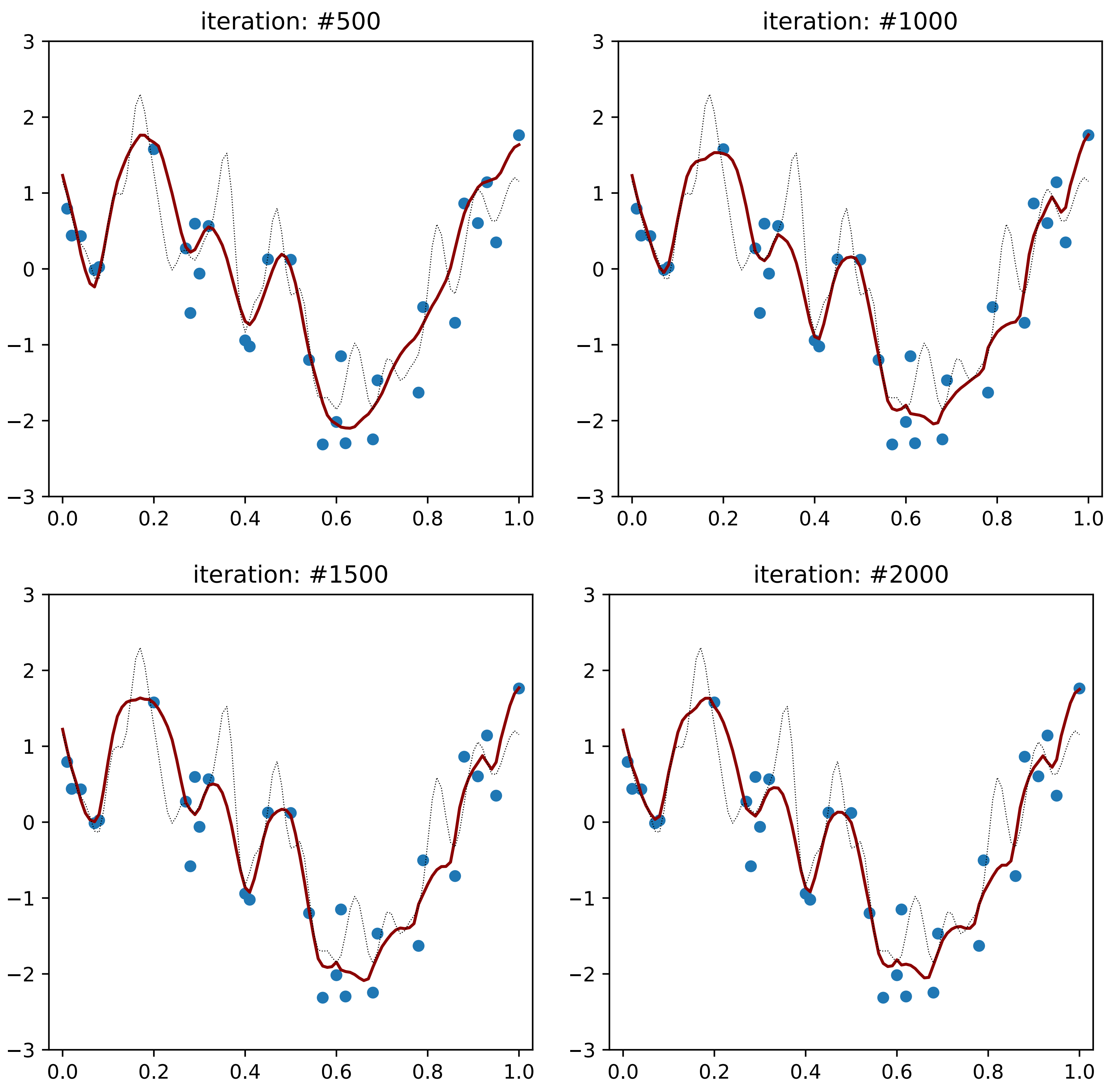
$$\mathrm{Diff}(\widetilde{T}) = \sum_{h=1}^{H} W_O^{(h)} \left(W_V^{(h)} \widetilde{T} \right) \left[\left(W_K^{(h)} \widetilde{T} \right)^{\top} \left(W_Q^{(h)} \widetilde{T} \right) \Big/ \sqrt{h_d} \right]$$

Intg is the cumulative summation operator.

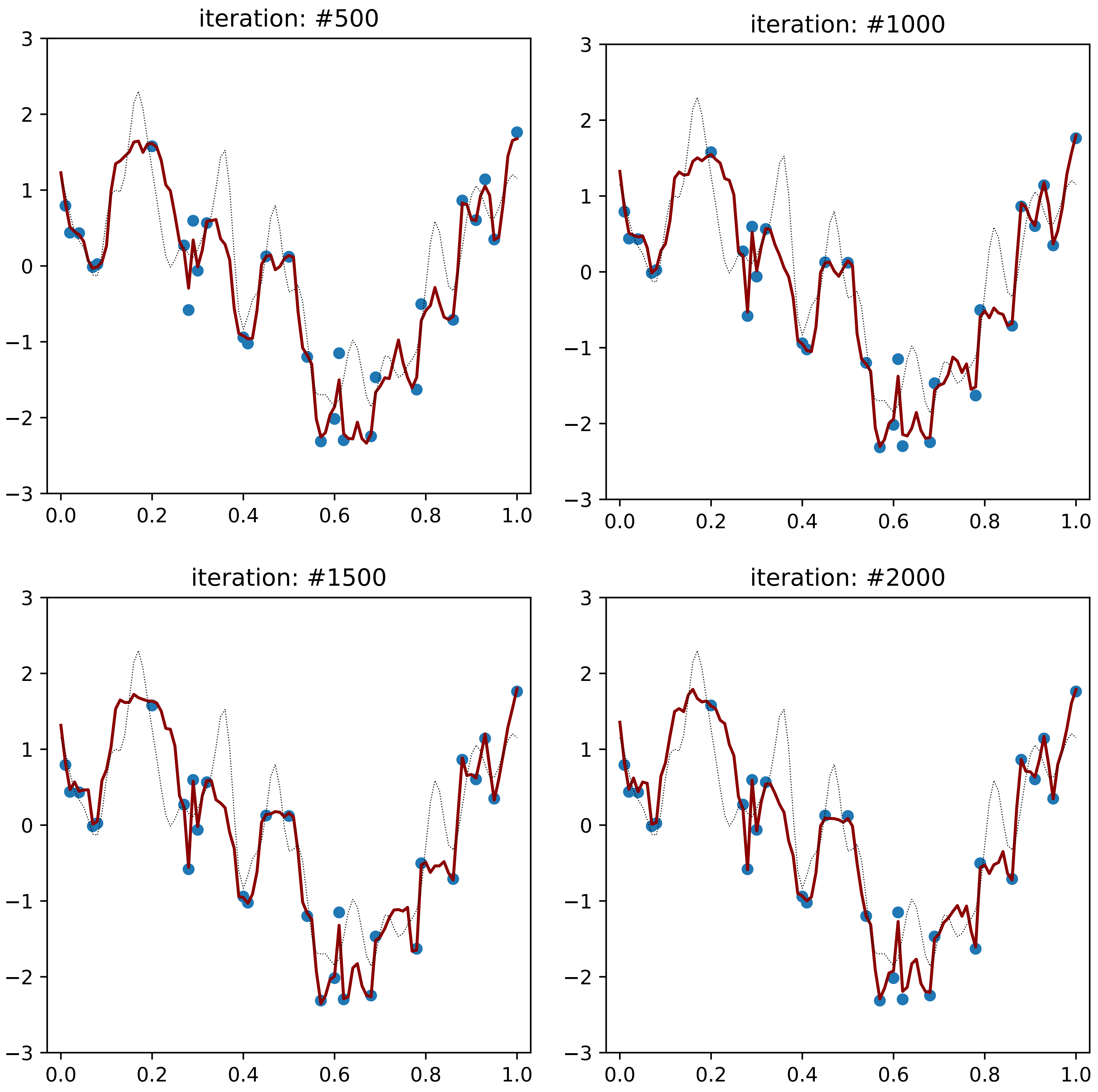
Output: a smooth version of an input

$$\mathrm{SAND}(\widetilde{T}) = (\widetilde{T})_1 + \mathrm{Intg}[\mathrm{Diff}(\widetilde{T})]$$

SAND — Compared to Vanilla Transformers



Imputation from SAND Over Iterations



Imputation from Vanilla Transformer Over Iterations

Simulation Studies

- Sample size $n = 10{,}000$. Signal-to-noise ratio $= 4$

	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SD)	TV(SD)	MSE(SD)	TV(SD)	MSE(SD)	TV(SD)
PACE[1]	189.9(4.3)	187.1(2.0)	450.0(15)	201.9(2.1)	795.5(33)	209.5(2.2)
FACE[5]	284.6(8.8)	198.9(2.1)	488.2(16)	204.5(2.2)	807.1(32)	209.5(2.2)
mFPCA[6]	224.7(5.8)	192.0(2.1)	480.3(16)	204.0(2.2)	787.1(31)	**209.3**(2.2)
MICE[7]	176.7(3.7)	233.1(1.7)	721.6(27)	318.4(3.0)	1416(57)	332.7(2.8)
CNP[2]	290.4(11)	198.9(2.0)	551.3(21)	207.6(2.1)	920.3(52)	211.9(2.2)
GAIN[8]	261.9(6.8)	350.0(3.4)	1767(52)	743.3(5.1)	2065(51)	759.2(4.3)
1DS	262.9(6.0)	273.8(2.4)	735.3(22)	305.7(3.7)	1157(43)	263.3(3.1)
Transformers and our method						
VT[3]	169.8(3.2)	218.2(1.7)	436.7(15)	227.0(2.2)	798.6(35)	230.6(2.6)
VTP	169.0(3.5)	179.9(2.0)	425.3(14)	**199.4**(2.1)	**777.4**(36)	210.2(2.2)
SAND	**146.5**(2.7)	**164.6**(1.8)	**410.9**(13)	**196.8**(2.0)	**758.1**(43)	**206.8**(2.2)

*MSE, TV: the smaller the better

Read Data

- Impute $n = 5500$ household's energy usage in London from Nov 13 — 14, 2013

	UK electricity					
	$n_i = 30$		$n_i = 8$ to 12		$n_i = 3, 4, 5$	
	MSE(SD)	TV(SD)	MSE(SD)	TV(SD)	MSE(SD)	TV(SD)
PACE	12.8(1.8)	**19.0**(1.1)	**30.1**(4.5)	**21.1**(1.2)	**39.6**(5.2)	**21.9**(1.2)
FACE	15.8(2.1)	21.3(1.2)	32.5(5.4)	22.6(1.2)	**39.6**(5.2)	23.0(1.2)
mFPCA	16.4(2.0)	22.2(1.2)	34.8(4.9)	23.2(1.2)	41.7(5.4)	23.3(1.2)
MICE	20.4(2.2)	67.8(3.3)	40.0(4.5)	65.4(2.8)	75.4(8.6)	71.4(1.5)
CNP	23.0(3.5)	21.4(1.2)	31.5(4.3)	22.1(1.2)	47.9(7.1)	**22.7**(1.2)
GAIN	31.9(3.7)	108(5.6)	75.4(8.2)	104(6.7)	99.6(15)	121(2.4)
1DS	17.3(2.2)	19.4(1.1)	50.0(7.0)	22.8(1.3)	105(18)	44.1(2.7)
VT	**10.7**(1.8)	20.6(1.1)	31.2(3.3)	23.2(1.3)	42.6(5.6)	38.5(2.5)
SAND	**10.0**(1.9)	**15.7**(0.9)	**26.7**(3.0)	**20.1**(1.2)	**38.3**(5.1)	25.5(1.6)

Reference

1. Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. Journal of the American Statistical Association, 2005.
2. Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In International Conference on Machine Learning, 2018.
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, 2017.
4. Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.
5. Luo Xiao, Cai Li, William Checkley, and Ciprian Crainiceanu. Fast covariance estimation for sparse functional data. Statistics and computing, 2018
6. Jie Peng and Debashis Paul. A geometric approach to maximum likelihood estimation of the functional principal components from sparse longitudinal data. Journal of Computational and Graphical Statistics, 2009
7. Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. Journal of Statistical Software, 2011.
8. Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In International conference on machine learning. Proceedings of Machine Learning Research, 2018.