Learning Neural Representations for Time Series

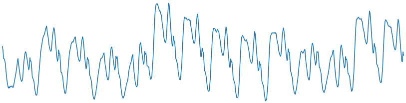
Defended by Etienne Le Naour

September 27, 2024

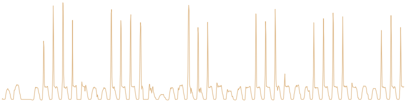
Committee members

Referees	Germain Forestier	Professor at Université de Haute-Alsace
	Romain Tavenard	Professor at Université de Rennes 2
Examiners	Isabelle Bloch	Professor at Sorbonne Université
	Marc Sebban	Professor at Université de Saint-Étienne
Supervisors	Vincent Guigue	Professor at AgroParisTech
	Nicolas Baskiotis	Associate Professor at Sorbonne Université
Invited guests	Ghislain Agoua	Research Scientist at EDF R&D
	Patrick Gallinari	Professor at Sorbonne Université

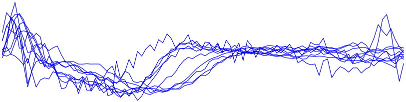
Time series are ubiquitous



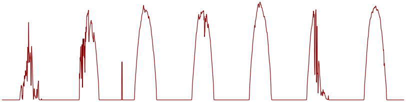
Aggregated load curve



Road occupancy



Heartbeat electrical activity

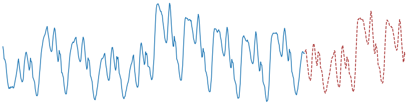


Solar power generation

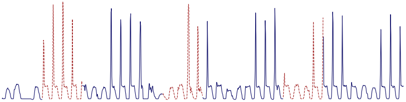
Time series definition

Consider a temporal phenomenon $\mathbf{x} \colon t \in \mathcal{T} \rightarrow \mathbf{x_t} \in \mathbb{R}^c$.
Given a sequence of observed timestamps $(t_1, t_2, \ldots, t_T)$, the time series x, can be defined as $x = (\mathbf{x_{t_1}}, \mathbf{x_{t_2}}, \ldots, \mathbf{x_{t_T}}) \in \mathbb{R}^{c \times T}$.

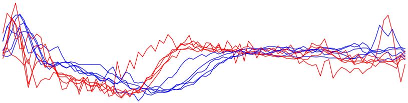
Time series are employed for different tasks



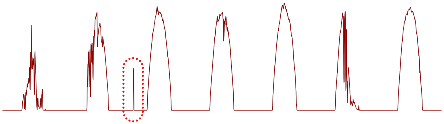
Forecasting



Imputation

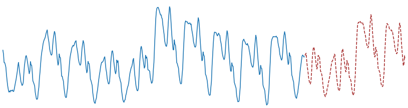


Classification

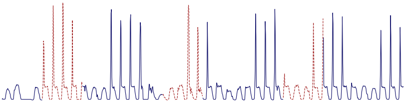


Anomaly detection

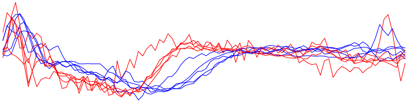
Time series are employed for different tasks



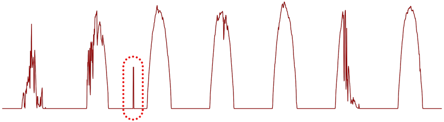
Forecasting



Imputation



Classification



Anomaly detection

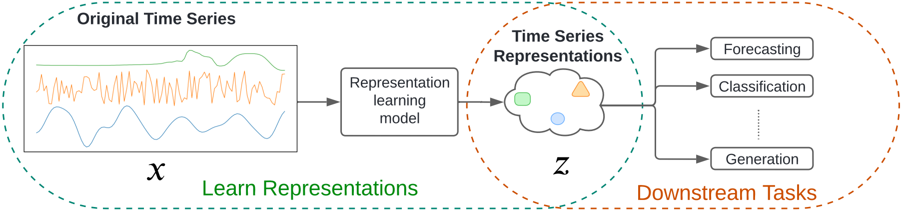
A long history of supervised models

- **Statistical** (Exponential Smoothing [Brown, 1959], ARIMA [Box et al., 1970])
- **Machine learning** (Shapelets [Ye and Keogh, 2011], Matrix factorization [Mei et al., 2017])
- **Deep learning** (InceptionTime [Ismail Fawaz et al., 2019], PatchTST [Nie et al., 2022])

Time series representation learning

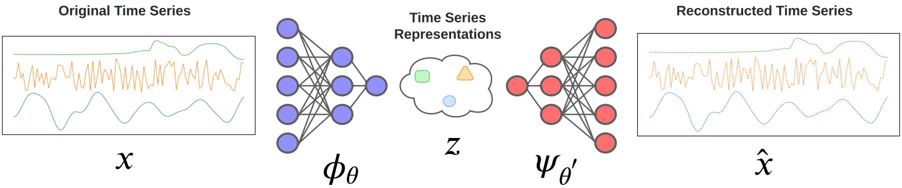
Time series representation definition

Given a time series $x^{(j)} \in \mathbb{R}^{c \times T}$, we define its representation as $z^{(j)} \in \mathbb{R}^{d \times T'}$, where $z^{(j)} = \phi(x^{(j)})$ and ϕ is an encoding function.



- Traditional time series representations rely on **pre-designed features extraction** (e.g. the seasonal / trend / residual decomposition [Taylor and Letham, 2017], the symbolic aggregate approximation [Lin et al., 2003]
- From traditional representations to neural representations

Time series neural representation



Wide range of models

- **Contrastive learning** [Franceschi et al., 2019, Yue et al., 2022]
- **Encoder-decoder based on reconstruction** [Nie et al., 2022, Zerveas et al., 2021]
- **Hybrid models** [Dong et al., 2024]

Leading to improvements

- Efficient classification for few-shot scenarios
- Create structured latent space
- Re-usability of the representation
- Transferable neural architecture

Research questions and industrial context

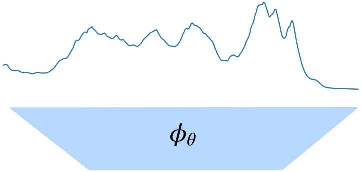
Some open problems

(i) **Interpretability of the neural representations**

(ii) Context adaptability through representation adjustment

(iii) Capture flexible representations

Existing time series neural representations are not interpretable



$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ \vdots \\ z_d \end{pmatrix}$$

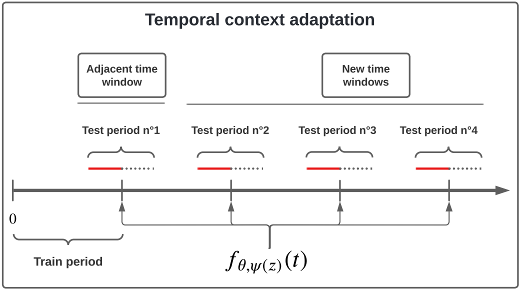
Research questions and industrial context

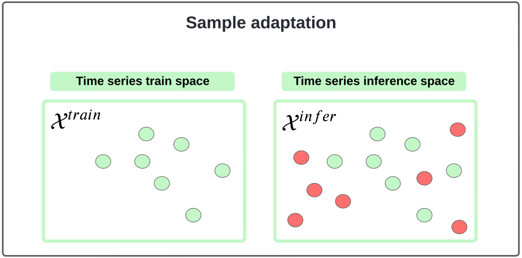
Some open problems

(i) Interpretability of the neural representations

(ii) **Context adaptability through representation adjustment**

(iii) Capture flexible representations





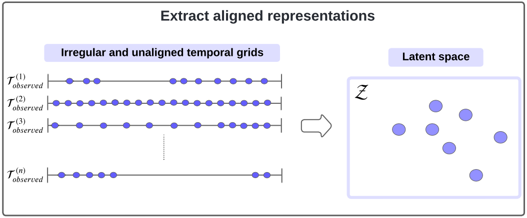
Research questions and industrial context

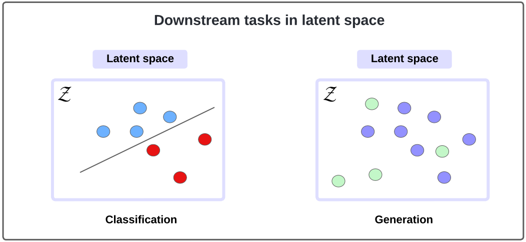
Some open problems

(i) Interpretability of the neural representations

(ii) Context adaptability through representation adjustment

(iii) **Capture flexible representations**





Outline

Introduction
- Context
- Motivations

Contributions
- Interpretable neural representation [1]
- TimeFlow [2]
- Flexible representations

Conclusion
- Synthesis
- Perspectives

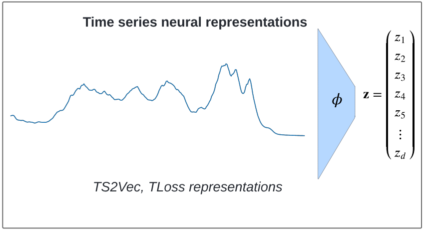
[1] *Interpretable time series neural representation for classification purposes*. IEEE International Conference on Data Science and Advanced Analytics 2023.

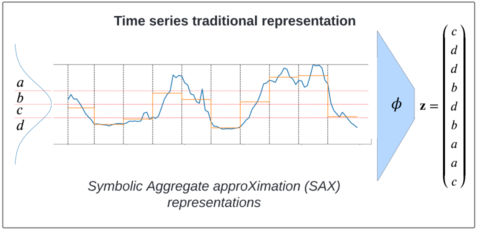
[2] *Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations*. Transactions on Machine Learning Research 2024.

Interpretable neural representation

Motivations

- Interpretability is key to decision maker adoption of deep learning
- Deep learning has advanced time series representation by identifying complex patterns but lacks interpretability
- Traditional representation learning methods offer interpretability, but fail to capture complex patterns





- Discrete neural representation is a promising option [van den Oord et al., 2017]

Requirements

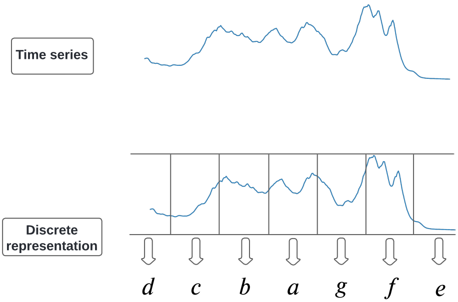
(i) **Discrete representation**
→ *Pair element with pattern*

(ii) Temporal consistency

(iii) A decodable representation

(iv) Shift equivariance property

(v) An adjustable representation



Requirements

(i) Discrete representation

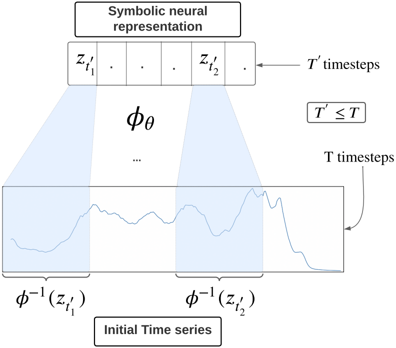
(ii) **Temporal consistency**

$\rightarrow$ *Localize elements*

(iii) A decodable representation

(iv) Shift equivariance property

(v) An adjustable representation



Requirements

(i) Discrete representation

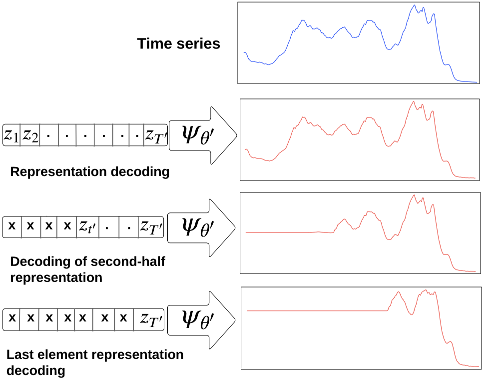
(ii) Temporal consistency

(iii) **A decodable representation**

$\rightarrow$ Localizable decoding

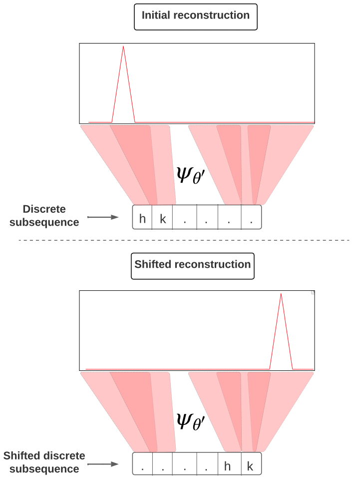
(iv) Shift equivariance property

(v) An adjustable representation



Requirements

- (i) Discrete representation
- (ii) Temporal consistency
- (iii) A decodable representation
- (iv) **Shift equivariance property**
  → Decode the same pattern with a shift in time
- (v) An adjustable representation



Requirements

(i) Discrete representation

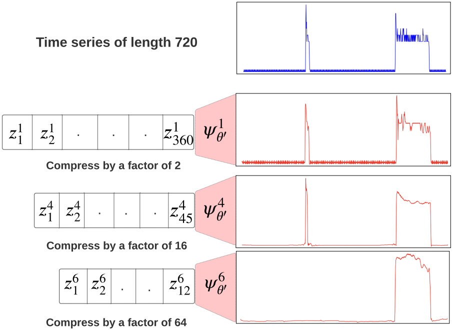
(ii) Temporal consistency

(iii) A decodable representation

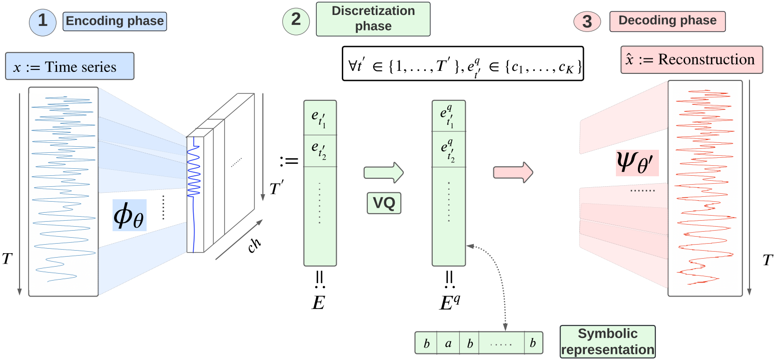
(iv) Shift equivariance property

(v) **An adjustable representation**

→ Control the amount of information captured



Proposed architecture

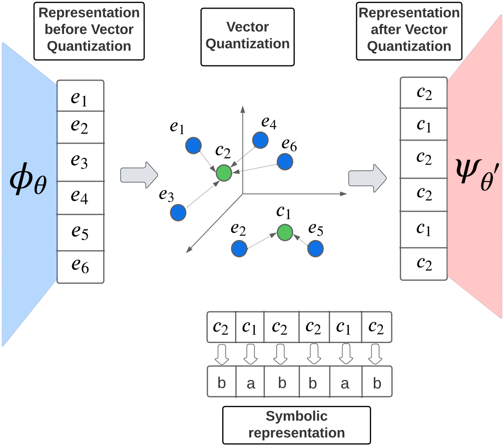


(i) Convolutional encoder with downsampling mechanisms

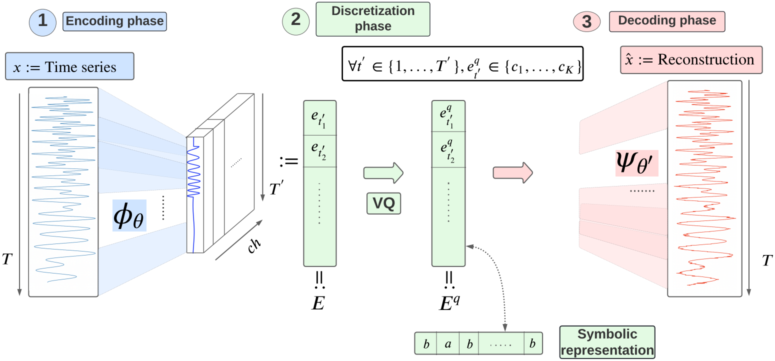
(ii) Vector Quantization (VQ) mechanism in the latent space

(iii) Convolutional decoder with upsampling mechanisms

Proposed architecture

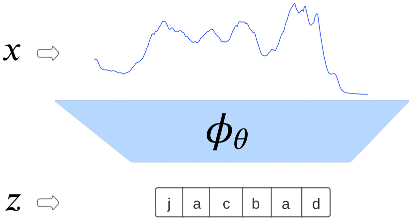


Proposed architecture



$$\underset{\theta,\theta',E}{\arg\min} \underbrace{||x-\psi_{\theta'}(E^q)||_2^2}_{\text{Reconstruction loss}} + \underbrace{||sg[\phi_\theta(x)]-E^q||_2^2}_{\text{Quantization loss}} + \beta \underbrace{||\phi_\theta(x)-sg[E^q]||_2^2}_{\text{Commitment loss}} .$$

Proposed architecture



- How to evaluate the usefulness of the learned representation $\mathbf{z}$?

Downstream task: classification over extracted representations

- From representation vector to features vector: $z \to h$
- Logistic regression on top of h

$$\mathbf{z} = \begin{pmatrix} a \\ j \\ a \\ b \\ b \\ \vdots \\ c \end{pmatrix} \quad \Longrightarrow \quad \mathbf{h} = \begin{pmatrix} \mathbb{1}_{a \in \mathbf{z}} \\ \mathbb{1}_{b \in \mathbf{z}} \\ \vdots \\ \mathbb{1}_{aa \in \mathbf{z}} \\ \mathbb{1}_{bb \in \mathbf{z}} \\ \vdots \end{pmatrix}$$

Extract fixed size feature vector from n-grams

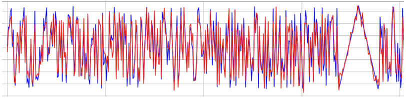
$$\arg\min_{w,b} \frac{1-\rho}{2} w^T w + \rho \|w\|_1 + \lambda \sum_{j=1}^{n} \log\left(\exp\left(-y_j \left({h^{(j)}}^T w + b\right)\right) + 1\right).$$

- We want a strong ℓ_1 penalty to ensure sparse feature selection

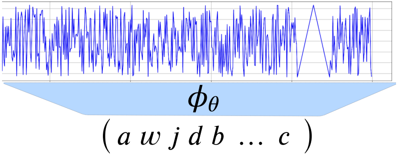
Qualitative experiment

Qualitative results

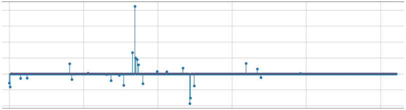
- *ShapeletSim dataset*
- Fit the unsupervised model and extract representations
- Classify and observe local and global interpretability



Train the VQ auto-encoder



Extract symbolic representations

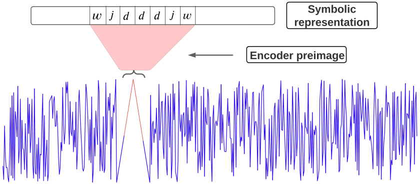


Classify and get features importance

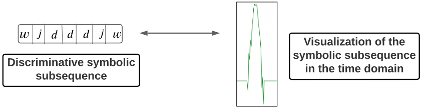
Qualitative experiment

Qualitative results

- *ShapeletSim dataset*
- Fit the unsupervised model and extract representations
- Classify and observe local and global interpretability



Local interpretability



Global interpretability

Quantitative experiments

Quantitative results

- Experiments performed on 25 datasets from the UCR archive
- Comparison to other interpretable methods
- Use only subsequences of length 2 or less

Datasets	Ours	SAX SEQL	SAX VSM	FS	LTS	1-NN DTW
Coffee	0.964	**1.000**	0.929	0.929	**1.000**	**1.000**
Computers	**0.728**	0.676	0.620	0.500	0.584	0.620
DistalPhalanxOAG	0.755	0.818	**0.842**	0.655	0.779	0.626
DistalPhalanxOC	0.732	0.718	0.728	**0.750**	0.719	0.725
DistalPhalanxTW	0.640	**0.748**	0.604	0.626	0.626	0.633
Earthquakes	0.734	**0.789**	0.748	0.705	0.741	0.727
ECG5000	**0.932**	0.924	0.910	0.923	**0.932**	0.925
FordA	0.883	0.851	0.827	0.787	**0.957**	0.691
GunPoint	0.940	0.987	0.987	0.947	**1.000**	0.913
Ham	0.705	0.705	**0.810**	0.648	0.667	0.600
Herring	**0.656**	0.578	0.625	0.531	0.625	0.531
ItalyPowerDemand	0.906	0.734	0.816	0.917	**0.970**	0.955
LargeKitchenApp	0.864	0.760	**0.877**	0.560	0.701	0.795
PhalangesOC	0.748	0.717	0.710	0.744	**0.765**	0.761
ProximalPhalanxOC	0.818	0.818	0.828	0.804	**0.834**	0.790
ProximalPhalanxOAG	0.839	0.844	0.824	0.780	**0.849**	0.785
ProximalPhalanxTW	0.771	**0.792**	0.610	0.702	0.776	0.756
RefrigerationDevices	0.533	0.541	**0.653**	0.333	0.515	0.440
ScreenType	0.499	0.461	**0.512**	0.413	0.429	0.411
ShapeletSim	0.994	0.994	0.717	**1.000**	0.950	0.700
SmallKitchenApp	**0.795**	0.776	0.579	0.333	0.664	0.672
Strawberry	**0.962**	0.954	0.957	0.903	0.911	0.946
Wafer	0.975	0.993	**0.999**	0.997	0.996	0.995
Wine	0.759	0.556	**0.963**	0.759	0.500	0.611
Worms	**0.714**	0.536	0.558	0.649	0.610	0.532
Mean	**0.793**	0.770	0.769	0.715	0.764	0.725
Win/Tie/Loss	/	**13**/3/9	**15**/0/10	**19**/1/5	**13**/1/11	**21**/0/4

Conclusion

Synthesis and perspectives

(i) Great interest for applications where they are few discriminative patterns

(ii) Possibility to greatly improve accuracy by explore longer subsequences for classification [Ifrim and Wiuf, 2011, Nguyen et al., 2019]

Limitations

(i) Rely heavily on some hyperparameters

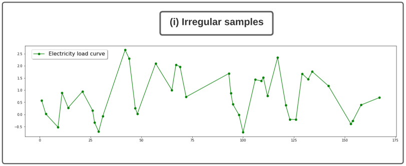
(ii) Less accurate than supervised SOTA classifiers

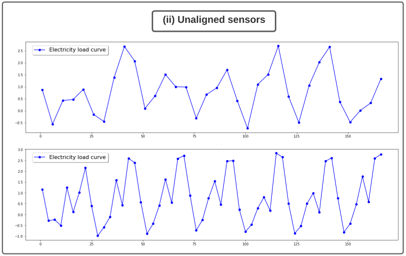
(iii) A framework limited to univariate time series

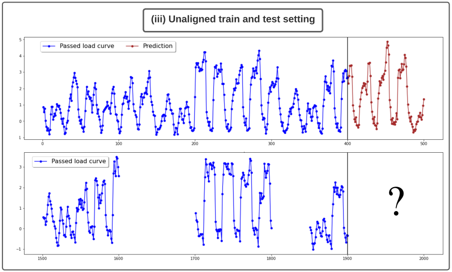
Le Naour, E., Agoua, G., Baskiotis, N., and Guigue, V. **Interpretable time series neural representation for classification purposes.** *IEEE 10th International Conference on Data Science and Advanced Analytics (IEEE DSAA)* 2023. Best research paper award.

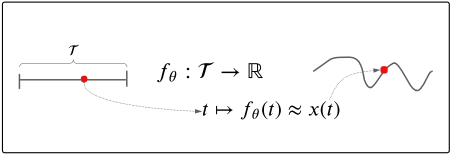
TimeFlow

Motivations: many scenarios require continuous modeling



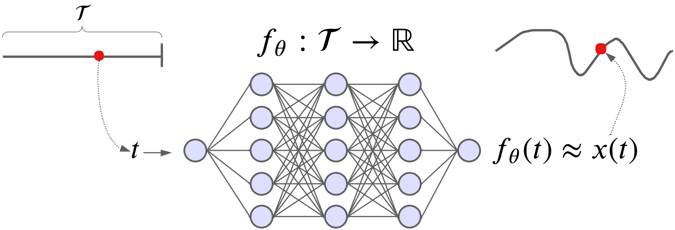




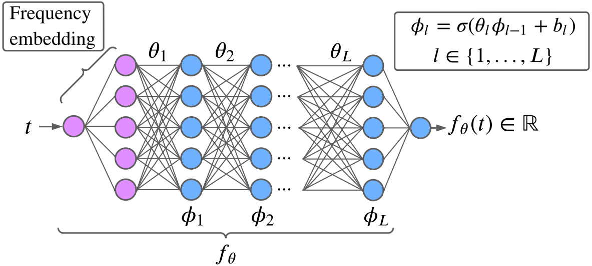


- Possible options : **Gaussian Processes** [Rasmussen and Williams, 2006], **Neural Processes** [Garnelo et al., 2018], **Implicit Neural Representations** [Sitzmann et al., 2020]

Implicit Neural Representations (INRs)



Implicit Neural Representations (INRs)

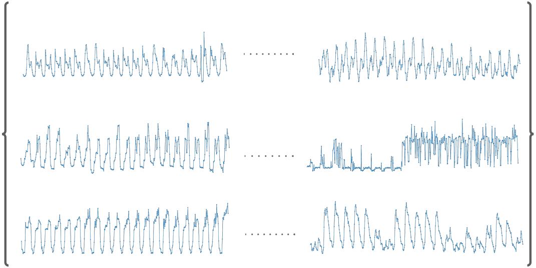


(i) NeRF encoding [Mildenhall et al., 2021] :
$t \mapsto \gamma(t) := (\sin(\pi t), \cos(\pi t), \cdots, \sin(2^N \pi t), \cos(2^N \pi t))$

(ii) $\gamma(t) \mapsto$ MLP$(\gamma(t); \theta)$ where $\sigma(\alpha) = \max(0, \alpha)$

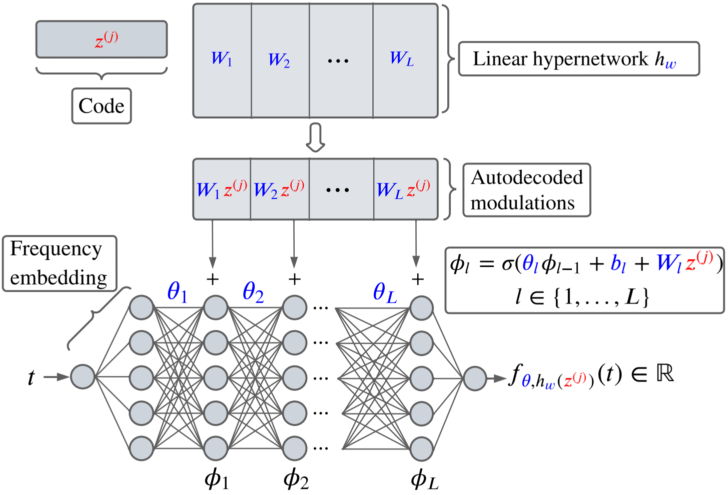
How to deal with datasets?

- Original INRs are designed to fit a single instance
- How to fit a dataset where series have shared and individual features ?



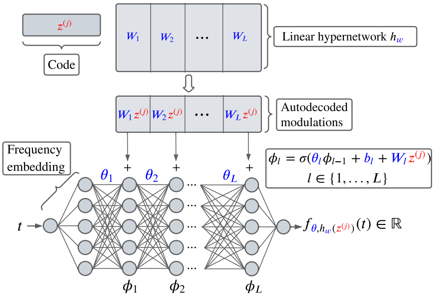
- A solution → **Hypernetwork that modulate the INR** [Dupont et al., 2022, Klocek et al., 2019]

Auto-decoding architecture



How to train parameters properly?

- **Metalearning** [Dupont et al., 2022, Serrano et al., 2023, Zintgraf et al., 2019]

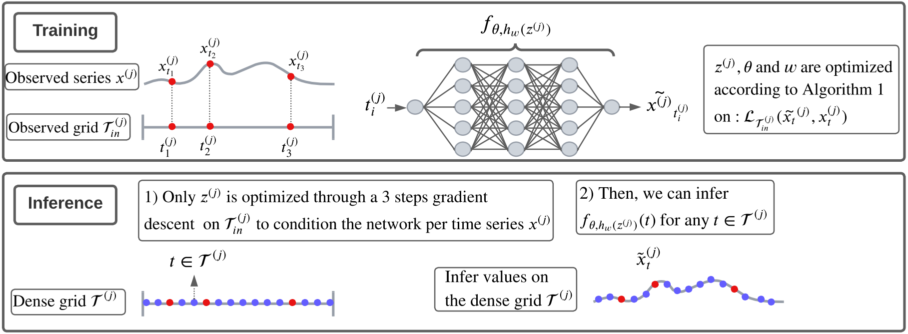


Algorithm 1: Training

while *no convergence* **do**
- Sample batch $\mathcal{B}$ of time series $(x^{(j)})_{j \in \mathcal{B}}$;
- Set codes to zero $z^{(j)} \leftarrow 0, \forall j \in \mathcal{B}$;
- `// inner loop for encoding:`
- **for** $j \in \mathcal{B}$ *and step* $\in \{1, \ldots, K\}$ **do**
  - $z^{(j)} \leftarrow z^{(j)} - \alpha \nabla_{z^{(j)}} \mathcal{L}_{\mathcal{T}}(f_{\theta, h_w(z^{(j)})}, x^{(j)})$;
- `// outer loop step:`
- $[\theta, w] \leftarrow [\theta, w] - \eta \nabla_{[\theta, w]} \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \mathcal{L}_{\mathcal{T}}(f_{\theta, h_w(z^{(j)})}, x^{(j)})$;

- The inner loop is responsible for encoding **individual context**
- E.g. **in forecasting**, $z^{(j)}$ is optimized on $\mathcal{T}_{in}^{(j)} \subset \mathcal{T}^{(j)}$

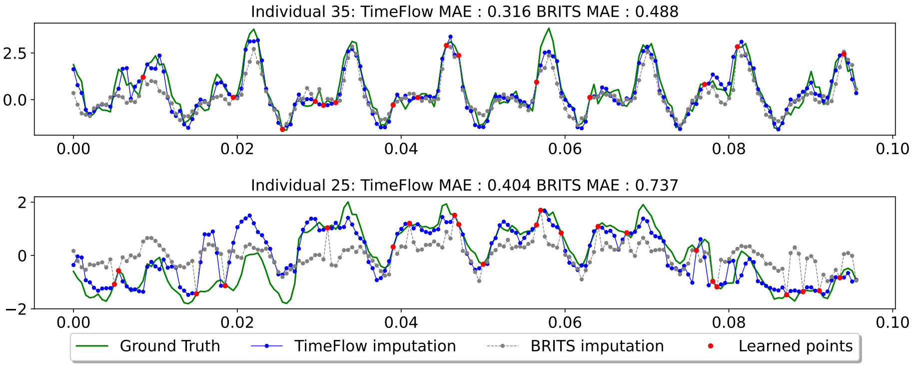
Imputation


Training
Observed series $x^{(j)}$
Observed grid $\mathcal{T}_{in}^{(j)}$
$x_{t_1}^{(j)}$ $x_{t_2}^{(j)}$ $x_{t_3}^{(j)}$
$t_1^{(j)}$ $t_2^{(j)}$ $t_3^{(j)}$
$f_{\theta,h_w(z^{(j)})}$
$t_i^{(j)}$
$\tilde{x}^{(j)}_{t_i^{(j)}}$
$z^{(j)}$, θ and w are optimized according to Algorithm 1 on : $\mathcal{L}_{\mathcal{T}_{in}^{(j)}}(\tilde{x}_t^{(j)}, x_t^{(j)})$
Inference
1) Only $z^{(j)}$ is optimized through a 3 steps gradient descent on $\mathcal{T}_{in}^{(j)}$ to condition the network per time series $x^{(j)}$
2) Then, we can infer $f_{\theta,h_w(z^{(j)})}(t)$ for any $t \in \mathcal{T}^{(j)}$
$t \in \mathcal{T}^{(j)}$
Dense grid $\mathcal{T}^{(j)}$
Infer values on the dense grid $\mathcal{T}^{(j)}$
$\tilde{x}_t^{(j)}$

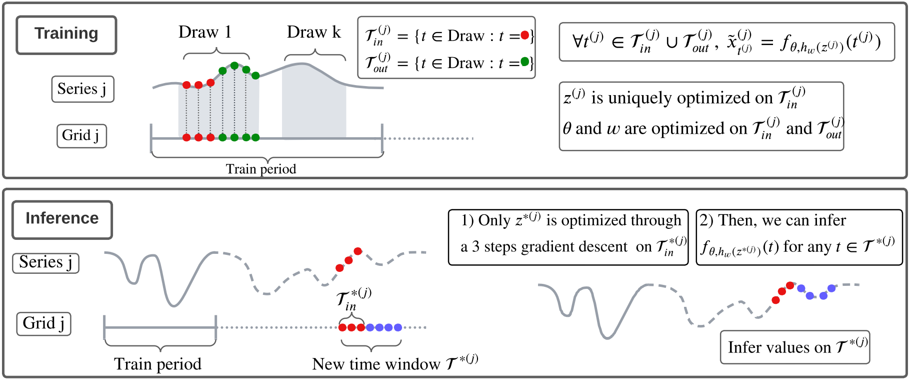
Imputation

		Continuous methods				Discrete methods			
	τ	TimeFlow	DeepTime	mTAN	Neural Process	CSDI	SAITS	BRITS	TIDER
Electricity	0.05	**0.324 ± 0.013**	0.379 ± 0.037	0.575 ± 0.039	0.357 ± 0.015	0.462 ± 0.021	0.384 ± 0.019	0.329 ± 0.015	0.427 ± 0.010
	0.10	**0.250 ± 0.010**	0.333 ± 0.034	0.412 ± 0.047	0.417 ± 0.057	0.398 ± 0.072	0.308 ± 0.011	0.287 ± 0.015	0.399 ± 0.009
	0.20	**0.225 ± 0.008**	0.244 ± 0.013	0.342 ± 0.014	0.320 ± 0.017	0.341 ± 0.068	0.261 ± 0.008	0.245 ± 0.011	0.391 ± 0.010
	0.30	**0.212 ± 0.007**	0.240 ± 0.014	0.335 ± 0.015	0.300 ± 0.022	0.277 ± 0.059	0.236 ± 0.008	0.221 ± 0.008	0.384 ± 0.009
	0.50	0.194 ± 0.007	0.227 ± 0.012	0.340 ± 0.022	0.297 ± 0.016	**0.168 ± 0.003**	0.209 ± 0.008	0.193 ± 0.008	0.386 ± 0.009
Solar	0.05	**0.095 ± 0.015**	0.190 ± 0.020	0.241 ± 0.102	0.115 ± 0.015	0.374 ± 0.033	0.142 ± 0.016	0.165 ± 0.014	0.291 ± 0.009
	0.10	**0.083 ± 0.015**	0.159 ± 0.013	0.251 ± 0.081	0.114 ± 0.014	0.375 ± 0.038	0.124 ± 0.018	0.132 ± 0.015	0.276 ± 0.010
	0.20	**0.072 ± 0.015**	0.149 ± 0.020	0.314 ± 0.035	0.109 ± 0.016	0.217 ± 0.023	0.108 ± 0.014	0.109 ± 0.012	0.270 ± 0.010
	0.30	**0.061 ± 0.012**	0.135 ± 0.014	0.338 ± 0.05	0.108 ± 0.016	0.156 ± 0.002	0.100 ± 0.015	0.098 ± 0.012	0.266 ± 0.010
	0.50	**0.054 ± 0.013**	0.098 ± 0.013	0.315 ± 0.080	0.107 ± 0.015	0.079 ± 0.011	0.094 ± 0.013	0.088 ± 0.013	0.262 ± 0.009
Traffic	0.05	0.283 ± 0.016	**0.246 ± 0.010**	0.406 ± 0.074	0.318 ± 0.014	0.337 ± 0.045	0.293 ± 0.007	0.261 ± 0.010	0.363 ± 0.007
	0.10	**0.211 ± 0.012**	0.214 ± 0.007	0.319 ± 0.025	0.288 ± 0.018	0.288 ± 0.017	0.237 ± 0.006	0.245 ± 0.009	0.362 ± 0.006
	0.20	**0.168 ± 0.006**	0.216 ± 0.006	0.270 ± 0.012	0.271 ± 0.011	0.269 ± 0.017	0.197 ± 0.005	0.224 ± 0.008	0.361 ± 0.006
	0.30	**0.151 ± 0.007**	0.172 ± 0.008	0.251 ± 0.006	0.259 ± 0.012	0.240 ± 0.037	0.180 ± 0.006	0.197 ± 0.007	0.355 ± 0.006
	0.50	**0.139 ± 0.007**	0.171 ± 0.005	0.278 ± 0.040	0.240 ± 0.021	0.144 ± 0.022	0.160 ± 0.008	0.161 ± 0.060	0.354 ± 0.007
TimeFlow improvement		/	24.14 %	50.53 %	31.61 %	36.12 %	20.33 %	18.90 %	53.40 %

Imputation


Individual 35: TimeFlow MAE : 0.316 BRITS MAE : 0.488
2.5
0.0
0.00
0.02
0.04
0.06
0.08
0.10
Individual 25: TimeFlow MAE : 0.404 BRITS MAE : 0.737
2
0
−2
Ground Truth
TimeFlow imputation
BRITS imputation
Learned points

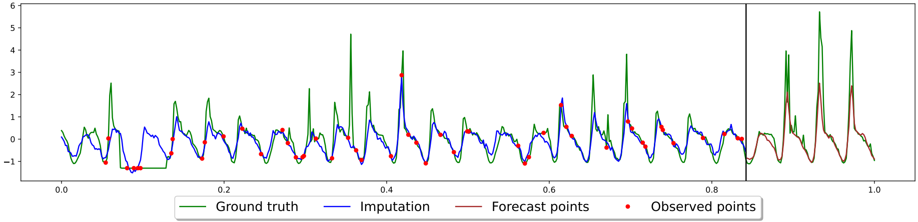
Forecasting



Forecasting

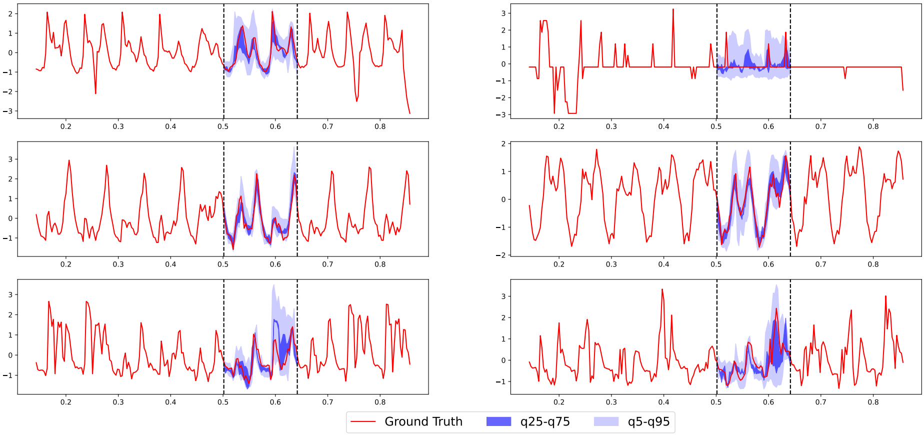
	H	Continuous methods			Discrete methods			
		TimeFlow	DeepTime	Neural Process	Patch-TST	DLinear	AutoFormer	Informer
Electricity	96	<u>0.218 ± 0.017</u>	0.240 ± 0.027	0.392 ± 0.045	**0.214 ± 0.020**	0.236 ± 0.035	0.310 ± 0.031	0.293 ± 0.0184
	192	<u>0.238 ± 0.012</u>	0.251 ± 0.023	0.401 ± 0.046	**0.225 ± 0.017**	0.248 ± 0.032	0.322 ± 0.046	0.336 ± 0.032
	336	<u>0.265 ± 0.036</u>	0.290 ± 0.034	0.434 ± 0.075	**0.242 ± 0.024**	0.284 ± 0.043	0.330 ± 0.019	0.405 ± 0.044
	720	<u>0.318 ± 0.073</u>	0.356 ± 0.060	0.605 ± 0.149	**0.291 ± 0.040**	0.370 ± 0.086	0.456 ± 0.052	0.489 ± 0.072
SolarH	96	**0.172 ± 0.017**	<u>0.197 ± 0.002</u>	0.221 ± 0.048	0.232 ± 0.008	0.204 ± 0.002	0.261 ± 0.053	0.273 ± 0.023
	192	**0.198 ± 0.010**	<u>0.202 ± 0.014</u>	0.244 ± 0.048	0.231 ± 0.027	0.211 ± 0.012	0.312 ± 0.085	0.256 ± 0.026
	336	<u>0.207 ± 0.019</u>	**0.200 ± 0.012**	0.241 ± 0.005	0.254 ± 0.048	0.212 ± 0.019	0.341 ± 0.107	0.287 ± 0.006
	720	**0.215 ± 0.016**	<u>0.240 ± 0.011</u>	0.403 ± 0.147	0.271 ± 0.036	0.246 ± 0.015	0.368 ± 0.006	0.341 ± 0.049
Traffic	96	<u>0.216 ± 0.033</u>	0.229 ± 0.032	0.283 ± 0.028	**0.201 ± 0.031**	0.225 ± 0.034	0.299 ± 0.080	0.324 ± 0.113
	192	<u>0.208 ± 0.021</u>	0.220 ± 0.020	0.292 ± 0.023	**0.195 ± 0.024**	0.215 ± 0.022	0.320 ± 0.036	0.321 ± 0.052
	336	<u>0.237 ± 0.040</u>	0.247 ± 0.033	0.305 ± 0.039	**0.220 ± 0.036**	0.244 ± 0.035	0.450 ± 0.127	0.394 ± 0.066
	720	**0.266 ± 0.048**	0.290 ± 0.045	0.339 ± 0.037	<u>0.268 ± 0.050</u>	0.290 ± 0.047	0.630 ± 0.043	0.441 ± 0.055
TimeFlow improvement		/	6.56 %	30.79 %	2.64 %	7.30 %	35.43 %	33.07 %

A flexible framework



Forecast on sparsely observed look-back window

A flexible framework



Quantifying uncertainty on block imputation ($\mathcal{L}$ is the pinball loss)

Conclusion

Synthesis

(i) A versatile framework that can handle a wide range of tasks and settings

(ii) Superior or comparable performance with discrete state-of-the-art baselines in imputation and forecasting

Limitations

(i) Slower at inference compare to existing baselines

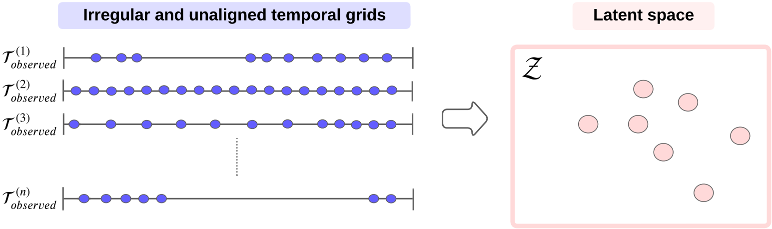
(ii) Does not allow drastic distribution shifts

Le Naour, E. *, Serrano, L. *, Migus, L. *, Yin, Y., Agoua, G., Baskiotis, N., Gallinari, P., and Guigue, V. **Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations.** *Transactions on Machine Learning Research (TMLR)* 2024.

Flexible representations

Learn representations from irregular/unaligned time series

Extract aligned representations



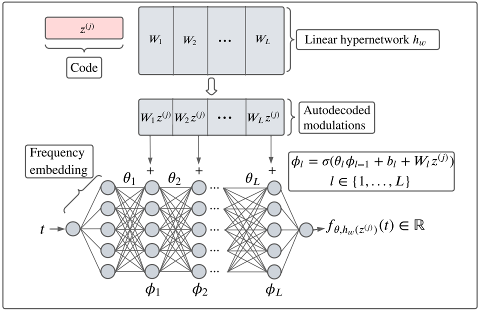
Motivations

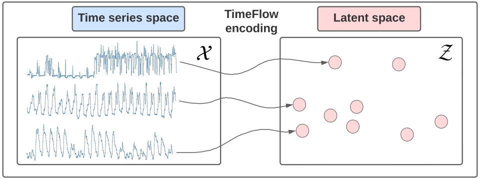
- A well-aligned latent space makes it easier to perform downstream tasks
- This two-stage approach is underexplored

TimeFlow can capture flexible representations

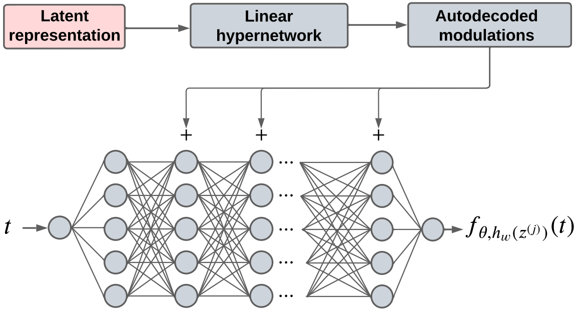
- Auto-decoding mechanism extracts a representation $z^{(j)} \in \mathbb{R}^d$ for a time series $x^{(j)} \in \mathbb{R}^{T^{(j)}}$
- For a time series dataset $\{\mathbf{x}^{(j)}\}_{j=1}^n$ we can build the corresponding representations dataset $\{\mathbf{z}^{(j)}\}_{j=1}^n$

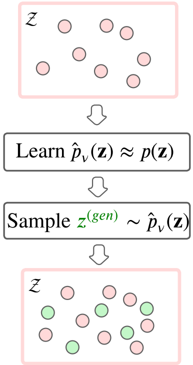
- Is the latent space $\mathcal{Z}$ efficient for downstream tasks ?





Example of downstream task : unconditional generation





- **Motivations**: data augmentation, overcoming privacy/property constraints
- **Training procedure**: (i) Fit TimeFlow (ii) Learn a Denoising Diffusion Probabilistic Model (DDPM) on the learned representations
- **Inference procedure**: (i) Sample a new representation (ii) Decode the representation

Experiments

Experimental setup

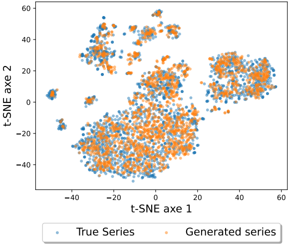
- Training on 8000 hourly time series (two-weeks long) from *Electricity*
- 2000 time series for testing and 2000 generated time series
- We compare with two baselines : DDPM only and TimeGan [Yoon et al., 2019]
- We want to assess the **fidelity** and diversity

	TimeFlow + DDPM	DDPM only	TimeGAN	Fully separable generation
Discriminative score ↓	**0.1388**	0.1704	0.4890	0.5000

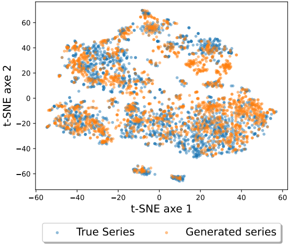
Experiments

Experimental setup

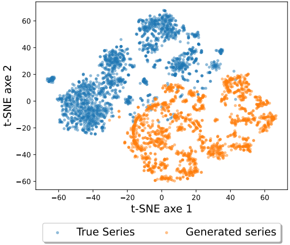
- Training on 8000 hourly time series (two-weeks long) from *Electricity*
- 2000 time series for testing and 2000 generated time series
- We compare with two baselines : DDPM only and TimeGan [Yoon et al., 2019]
- We want to assess the fidelity and **diversity**



(a) t-SNE ours



(b) t-SNE DDPM only



(c) t-SNE TimeGan

Conclusion

Synthesis

(i) A semantically rich latent space

(ii) The representations can encode irregular/unaligned time series

(iii) First unconditional generation experiments are convincing

Limitations and perspectives

(i) Unconditional generation experiments performed on only one dataset

(ii) Other downstream tasks should be explored to assess usefulness for downstream tasks

Conclusion

Synthesis

Some open problems

(i) **Interpretability of the neural representations**
(ii) Context adaptability through representation adjustment
(iii) Capture flexible representations

Contributions

(i) **Requirements + Discrete neural representation (VQ-AE)**
(ii) Continuous modeling coupled with auto-decoding and meta-learning
(iii) TimeFlow representations

Synthesis

Some open problems

(i) Interpretability of the neural representations
(ii) **Context adaptability through representation adjustment**
(iii) Capture flexible representations

Contributions

(i) Requirements + Discrete neural representation (VQ-AE)
(ii) **Continuous modeling coupled with auto-decoding and meta-learning**
(iii) TimeFlow representations

Synthesis

Some open problems

(i) Interpretability of the neural representations

(ii) Context adaptability through representation adjustment

(iii) **Capture flexible representations**

Contributions

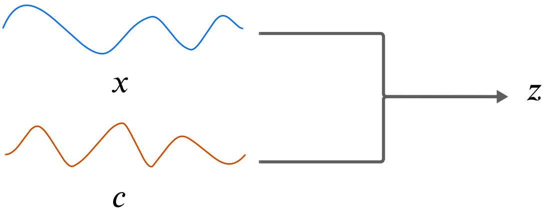
(i) Requirements + Discrete neural representation (VQ-AE)

(ii) Continuous modeling coupled with auto-decoding and meta-learning

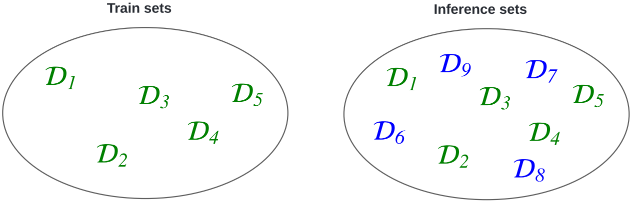
(iii) **TimeFlow representations**

Perspectives

Multivariate time series and context variables



Foundation models for time series



Thank you !

International publications

- Le Naour, E., Agoua, G., Baskiotis, N., and Guigue, V. **Interpretable time series neural representation for classification purposes.** *IEEE 10th International Conference on Data Science and Advanced Analytics (IEEE DSAA)* 2023. Best research paper award.
- Le Naour, E. *, Serrano, L. *, Migus, L. *, Yin, Y., Agoua, G., Baskiotis, N., Gallinari, P., and Guigue, V. **Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations.** *Transactions on Machine Learning Research (TMLR)* 2024.
- Keisler, J. *, Le Naour, E. * **WindDragon: Enhancing Wind Power Forecasting with Automated Deep Learning.** *International Conference on Learning Representations (ICLR), Tackling Climate Change with Machine Learning Workshop*, 2024.
- Serrano, L., Wang, T., Le Naour, E., Vittaut, J.N., Gallinari, P. **AROMA: Preserving Spatial Structure for Latent PDE Modeling with Local Neural Fields.** *Conference on Neural Information Processing Systems (Neurips)* 2024.

National publications

- Le Naour, E., Agoua, G., Baskiotis, N., and Guigue, V. **Représentation Interprétable pour la Classification de Séries Temporelles.** *Conférence d'Apprentissage Automatique (CAp)* 2023.
- Le Naour, E. *, Serrano, L. *, Migus, L. *, Yin, Y., Agoua, G., Baskiotis, N., Gallinari, P., and Guigue, V. **TimeFlow : Modélisation continue des séries temporelles avec représentations neuronales implicites.** *Conférence d'Apprentissage Automatique (CAp)* 2024.

---

* *Equal contribution.*

References i

References

G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 1970.

R. Brown. *Statistical Forecasting for Inventory Control*. McGraw-Hill, 1959. URL https://books.google.fr/books?id=oKI8AAAAIAAJ.

J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

E. Dupont, H. Kim, S. A. Eslami, D. J. Rezende, and D. Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022.

References ii

J. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems 32*, pages 4652–4663, 2019.

M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. M. A. Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80, pages 1690–1699. PMLR, 2018.

G. Ifrim and C. Wiuf. Bounded coordinate-descent for biological sequence classification in high dimensional predictor space. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 708–716. ACM, 2011. doi: 10.1145/2020408.2020519.

H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4): 917–963, 2019.

References iii

S. Klocek, Ł. Maziarka, M. Wołczyk, J. Tabor, J. Nowak, and M. Śmieja. Hypernetwork functional image representation. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28*, pages 496–510. Springer, 2019.

J. Lin, E. J. Keogh, S. Lonardi, and B. Y. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11, 2003. doi: 10.1145/882082.882086.

J. Mei, Y. de Castro, Y. Goude, and G. Hébrail. Nonnegative matrix factorization for time series recovery from a few temporal aggregates. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2382–2390, 2017.

References iv

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

T. L. Nguyen, S. Gsponer, I. Ilie, M. O'Reilly, and G. Ifrim. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data Min. Knowl. Discov.*, 33(4):1183–1222, 2019. doi: 10.1007/s10618-019-00633-3.

Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *CoRR*, abs/2211.14730, 2022.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

L. Serrano, L. L. Boudec, A. K. Koupaï, T. X. Wang, Y. Yin, J. Vittaut, and P. Gallinari. Operator learning with neural fields: Tackling pdes on general geometries. In *NeurIPS*, 2023.

References V

V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

S. J. Taylor and B. Letham. Forecasting at scale. *PeerJ Prepr.*, 5:e3190, 2017.

A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30*, pages 6306–6315, 2017.

L. Ye and E. J. Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Discov.*, 22(1-2):149–182, 2011. doi: 10.1007/s10618-010-0179-5.

J. Yoon, D. Jarrett, and M. Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. Ts2vec: Towards universal representation of time series. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 8980–8987. AAAI Press, 2022.

References vi

G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2114–2124. ACM, 2021. doi: 10.1145/3447548.3467401.

L. Zintgraf, K. Shiarli, V. Kurin, K. Hofmann, and S. Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.