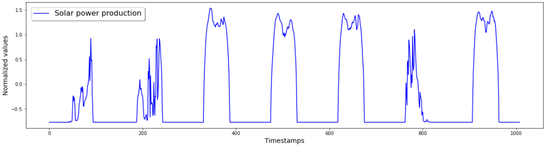
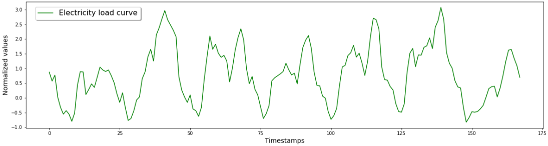
# An introduction to TimeFlow: Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations

July 2024

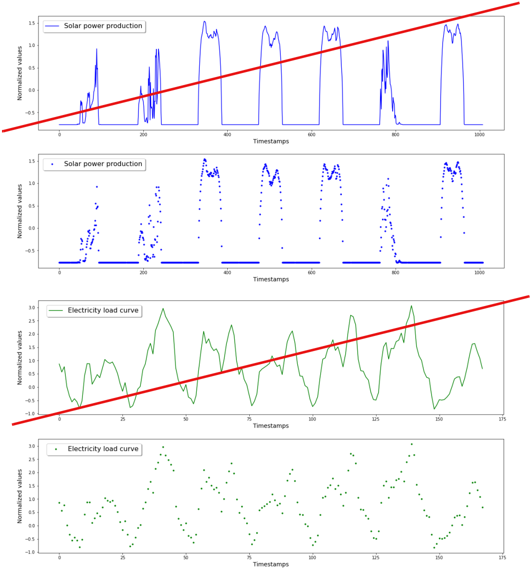
# Motivations

# Most measured phenomena in time series are continuous phenomena

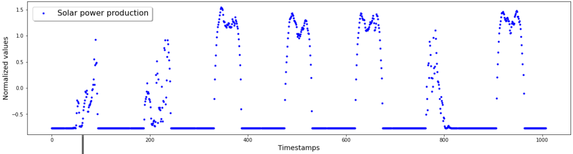


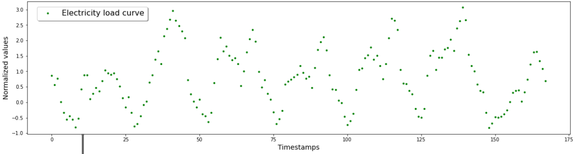
# But in real life, we observe partially these phenomena



# In a tabular way

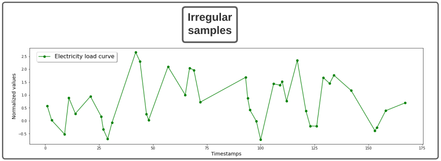


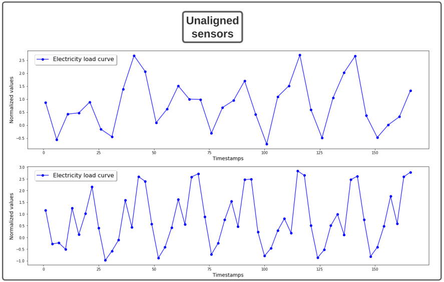

| Timestamps | 1 | 2 | 3 | 4 | 5 | | | | 1008 |
|---|---|---|---|---|---|---|---|---|---|
| Values | -0.77 | -0.77 | -0.77 | -0.77 | -0.77 | ... | ... | .... | -0.77 |

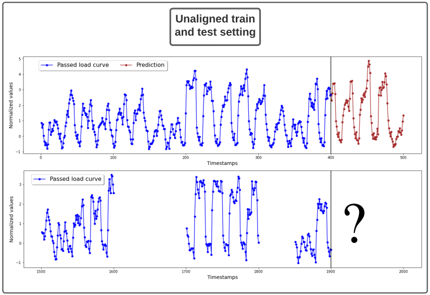



| Timestamps | 1 | 2 | 3 | 4 | 5 | | | | 168 |
|---|---|---|---|---|---|---|---|---|---|
| Values | 0.87 | 0.57 | 0.77 | 0.02 | -0.33 | ... | ... | .... | 0.69 |

# Tabular representation is convenient in machine learning but is limited for real-life problems

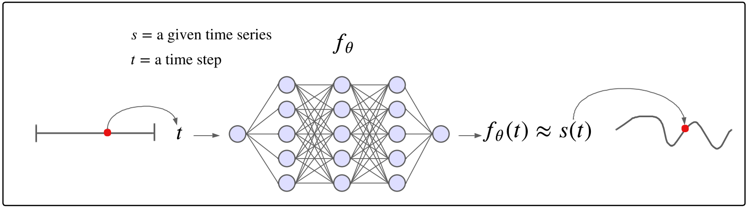


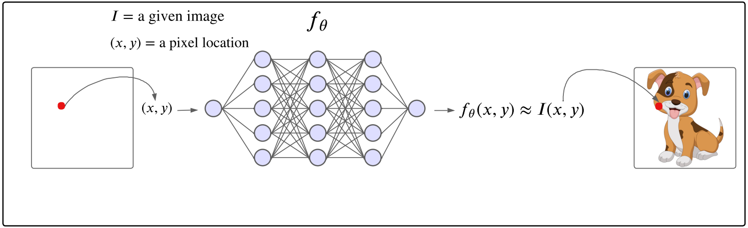



# Implicit Neural Representations (INRs)

# INRs treat structured data as a continuous function partially observed



# Let's drop the tabular representation for a continuous functional representation

## How to train $f_\theta$ ?

We solve the following optimization problem:

$$\theta^* = \arg\min_\theta \sum_{t \in \mathcal{T}^{obs}} \mathcal{L}(f_\theta(t), s(t))$$

- $\mathcal{T}^{obs}$ stands for the observed temporal support ($\mathcal{T}^{obs} \subset \mathcal{T}$)
- $\mathcal{L}$ stands for a differentiable loss (e.g. $\mathcal{L}(x, \tilde{x}) = ||x - \tilde{x}||^2$)

# How $f_\theta$ looks like

## NeRF encoding [Mildenhall et al., 2021]

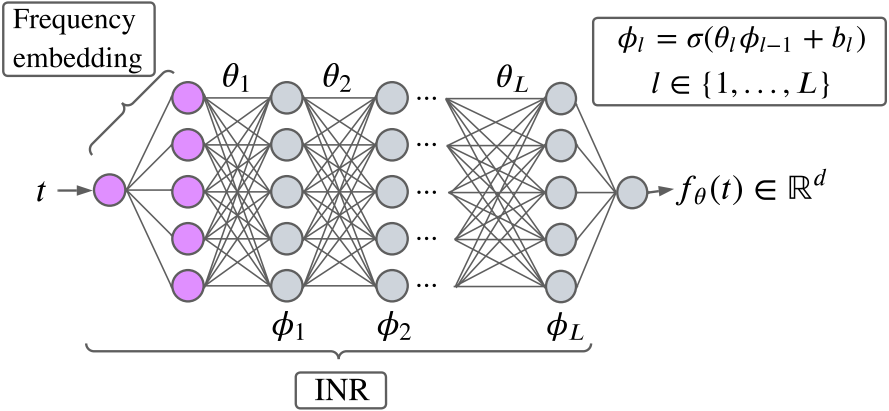
1. NeRF encoding : $t \rightarrow \gamma(t)$,
   $\gamma(t) := (\sin(\pi t), \cos(\pi t), \cdots, \sin(2^N \pi t), \cos(2^N \pi t))$
   • $N$ is the number of frequency bands
2. Then $\gamma(t) \rightarrow \mathrm{MLP}(\gamma(t); \theta)$
   Activation functions are ReLU (i.e. $ReLU(x) = max(0, x)$)

## SIREN approach [Sitzmann et al., 2020]

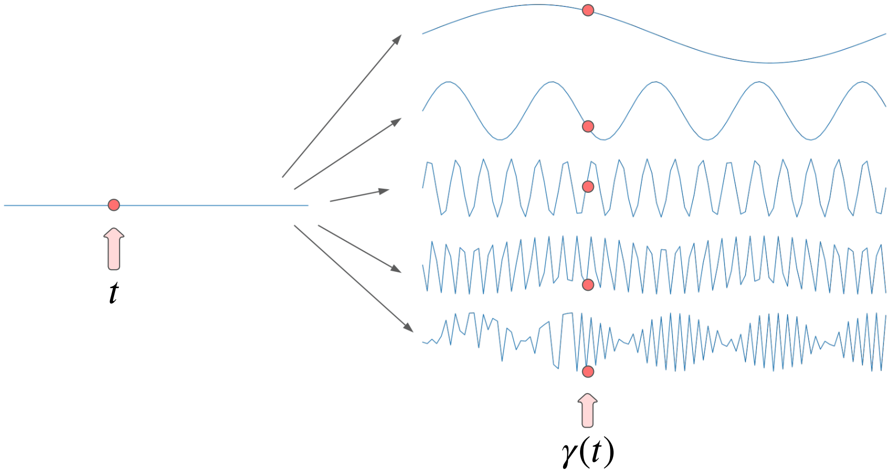
- $t \rightarrow \mathrm{MLP}(t; \theta)$, where layer l is:

$$A^{(l)} = sin(\omega_0 A^{(l-1)} W^{(l)} + b^{(l)})$$

# NeRF encoding illustration (1/2)

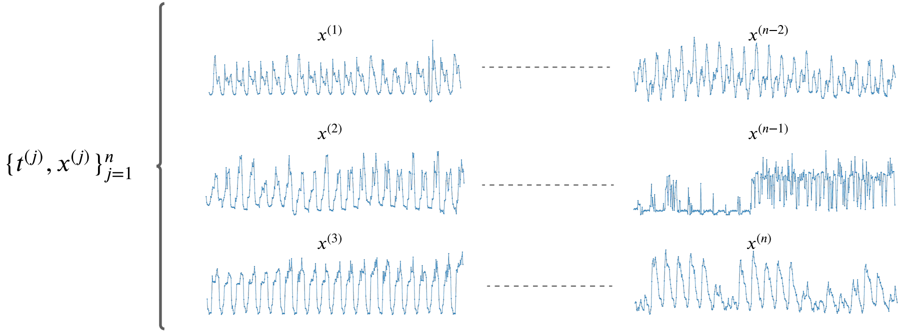


# NeRF encoding illustration (2/2)



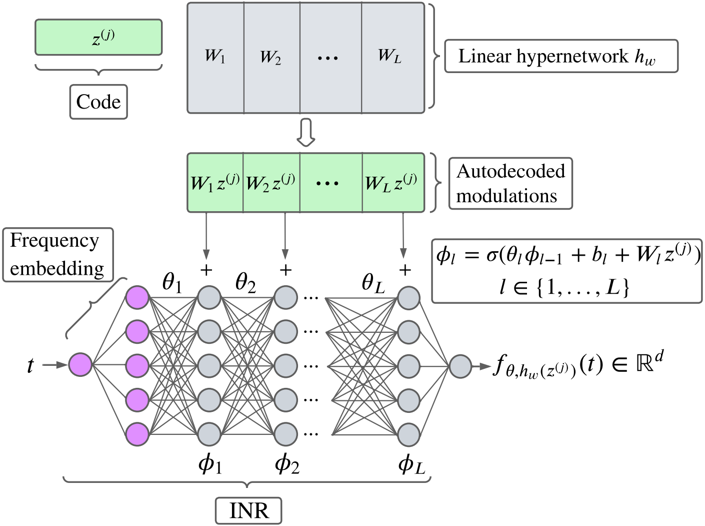
# TimeFlow

# Nice to fit a sample, but how to deal with a dataset?




- Solution $\rightarrow$ **Hypernetwork that modulate the INR** [Dupont et al., 2022, Klocek et al., 2019, Sitzmann et al., 2020]

# Hypernetwork and auto-decoding [Dupont et al., 2022, Yin et al., 2022]



# Insight on $\theta$, $w$ and the $z^{(j)}$

## New optimization problem

We want to retrieve $\theta^*, w^*, \begin{pmatrix} z^{(1)*} \\ \vdots \\ z^{(n)*} \end{pmatrix}$ that minimize:

$$\sum_{j \in \{1,\ldots,n\}} \sum_{t \in \mathcal{T}^{(j)obs}} \mathcal{L}(f_{\theta,w,z^{(j)}}(t), x^{(j)}(t))$$

- $\theta$ and $w$ are shared across all samples
- $z^{(j)}$ is only in relation to sample $j$.

## Key Concept

For each sample j, the parameter space is conditioned by $z^{(j)}$. Therefore, while $\theta$ and $w$ hold the shared information across all samples, the individual information is stored in $z^{(j)}$

# Training and meta-learning [Zintgraf et al., 2019]

**Algorithm 1:** Training

**while** *no convergence* **do**

Sample batch $\mathcal{B}$ of data $(x^{(j)})_{j\in\mathcal{B}}$;

Set codes to zero $z^{(j)} \leftarrow 0, \forall j \in \mathcal{B}$ ;

`// inner loop for encoding:`

**for** $j \in \mathcal{B}$ *and step* $\in \{1, ..., K\}$ **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:`

$[\theta, w] \leftarrow [\theta, w] - \eta \nabla_{[\theta, w]} \frac{1}{|\mathcal{B}|} \sum_{j\in\mathcal{B}} \mathcal{L}_{\mathcal{T}}(f_{\theta, h_w(z^{(j)})}, x^{(j)})$ ;

# At inference

**Algorithm 2:** Inference with trained $\theta, w$

For the $j$-th series $(x^{(j)})$, set code to zero $z^{*(j)} \leftarrow 0$;

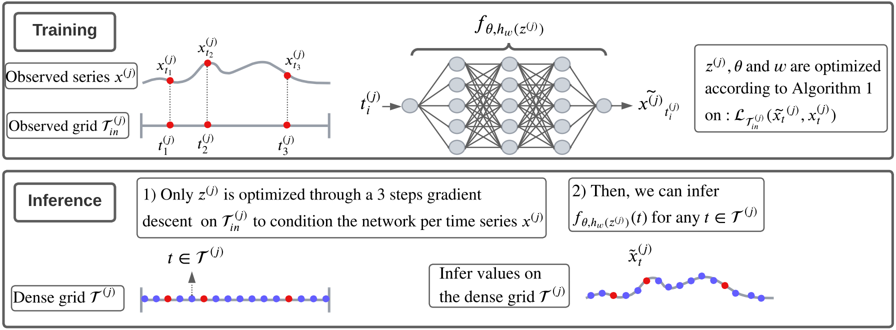
**for** $step \in \{1, \ldots, K\}$ **do**

$\quad z^{*(j)} \leftarrow z^{*(j)} - \alpha \nabla_{z^{*(j)}} \mathcal{L}_{\mathcal{T}}(f_{\theta, h_w(z^{*(j)})}, x_t)$

Query $f_{\theta, h_w(z^{*(j)})}(t)$ for any $t \in \mathcal{T}^{*(j)}$

# Experiments

# Imputation


Training
Observed series $x^{(j)}$
Observed grid $\mathcal{T}_{in}^{(j)}$
$x_{t_1}^{(j)}$
$x_{t_2}^{(j)}$
$x_{t_3}^{(j)}$
$t_1^{(j)}$
$t_2^{(j)}$
$t_3^{(j)}$
$f_{\theta,h_w(z^{(j)})}$
$t_i^{(j)}$
$\tilde{x}^{(j)}_{t_i^{(j)}}$
$z^{(j)}$, $\theta$ 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 $\mathcal{T}^{(j)}$
$\tilde{x}_t^{(j)}$

# We compare to a wide range of baselines on three datasets

Table: Mean MAE imputation results on the missing grid only. $\tau$ stands for the subsampling rate. Bold results are best, underlined results are second best.

| | | Continuous methods | | | | Discrete methods | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tau$ | 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 % |

# Qualitative comparison with BRITS

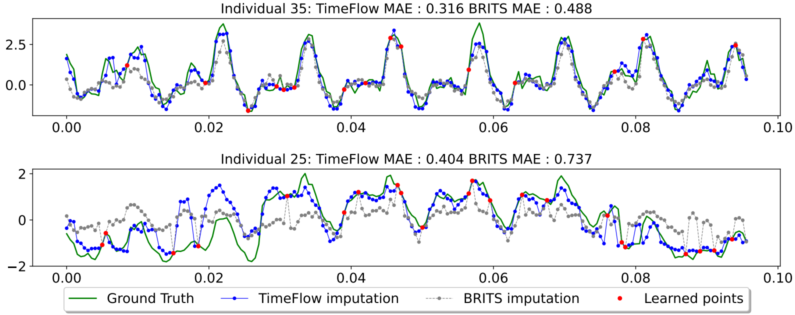
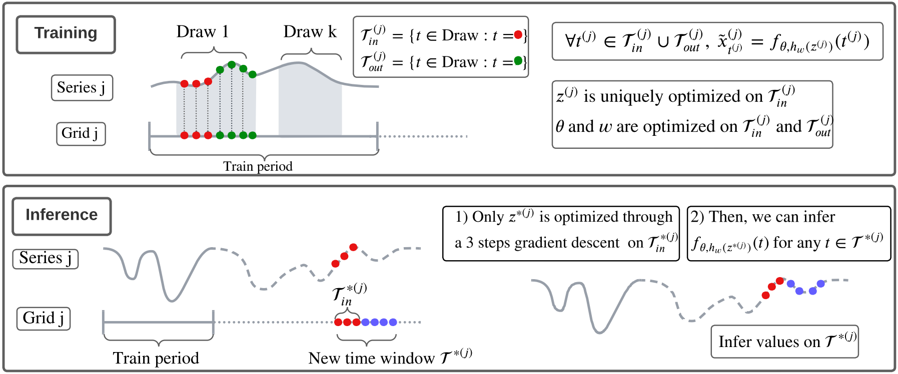



Figure: *Electricity dataset*. TimeFlow imputation (blue line) and BRITS imputation (gray line) with 10% of known point (red points) on the eight first days of samples 35 (top) and 25 (bottom).

# Forecasting



# We compare to a wide range of baselines on three datasets

Table: Mean MAE forecast results for adjacent time windows. H stands for the horizon. Bold results are best, underline results are second best.

| | | Continuous methods | | | Discrete methods | | | |
|---|---|---|---|---|---|---|---|---|
| | $H$ | 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 % |

# TimeFlow can even forecast on sparsely observed look-back window (1/2)

Table: MAE results for forecasting with missing values in the look-back window. $\tau$ stands for the percentage of observed values in the look-back window. Best results are in bold.

| | | | TimeFlow | | DeepTime | | Neural Process | |
|---|---|---|---|---|---|---|---|---|
| | $H$ | $\tau$ | Imputation error | Forecast error | Imputation error | Forecast error | Imputation error | Forecast error |
| Electricity | 96 | 0.5 | **0.151** ± **0.003** | **0.239** ± **0.013** | 0.209 ± 0.004 | 0.270 ± 0.019 | 0.460 ± 0.048 | 0.486 ± 0.078 |
| | | 0.2 | **0.208** ± **0.006** | **0.260** ± **0.015** | 0.249 ± 0.006 | 0.296 ± 0.023 | 0.644 ± 0.079 | 0.650 ± 0.095 |
| | | 0.1 | **0.272** ± **0.006** | **0.295** ± **0.016** | 0.284 ± 0.007 | 0.324 ± 0.026 | 0.740 ± 0.083 | 0.737 ± 0.106 |
| | 192 | 0.5 | **0.149** ± **0.004** | **0.235** ± **0.011** | 0.204 ± 0.004 | 0.265 ± 0.018 | 0.461 ± 0.045 | 0.498 ± 0.070 |
| | | 0.2 | **0.209** ± **0.006** | **0.257** ± **0.013** | 0.244 ± 0.007 | 0.290 ± 0.023 | 0.601 ± 0.075 | 0.626 ± 0.101 |
| | | 0.1 | **0.274** ± **0.010** | **0.289** ± **0.016** | 0.282 ± 0.007 | 0.315 ± 0.025 | 0.461 ± 0.045 | 0.724 ± 0.090 |
| Traffic | 96 | 0.5 | **0.180** ± **0.016** | **0.219** ± **0.026** | 0.272 ± 0.028 | 0.243 ± 0.030 | 0.436 ± 0.025 | 0.444 ± 0.047 |
| | | 0.2 | **0.239** ± **0.019** | **0.243** ± **0.027** | 0.335 ± 0.026 | 0.293 ± 0.027 | 0.596 ± 0.049 | 0.597 ± 0.075 |
| | | 0.1 | **0.312** ± **0.020** | **0.290** ± **0.027** | 0.385 ± 0.025 | 0.344 ± 0.027 | 0.734 ± 0.102 | 0.731 ± 0.132 |
| | 192 | 0.5 | **0.176** ± **0.014** | **0.217** ± **0.017** | 0.241 ± 0.027 | 0.234 ± 0.021 | 0.477 ± 0.042 | 0.476 ± 0.043 |
| | | 0.2 | **0.233** ± **0.017** | **0.236** ± **0.021** | 0.286 ± 0.027 | 0.276 ± 0.020 | 0.685 ± 0.109 | 0.678 ± 0.108 |
| | | 0.1 | **0.304** ± **0.019** | **0.277** ± **0.021** | 0.331 ± 0.025 | 0.324 ± 0.021 | 0.888 ± 0.178 | 0.877 ± 0.174 |
| TimeFlow improvement | | | / | / | 18.97 % | 11.87 % | 61.88 % | 58.41 % |

# TimeFlow can even forecast on sparsely observed look-back window (2/2)

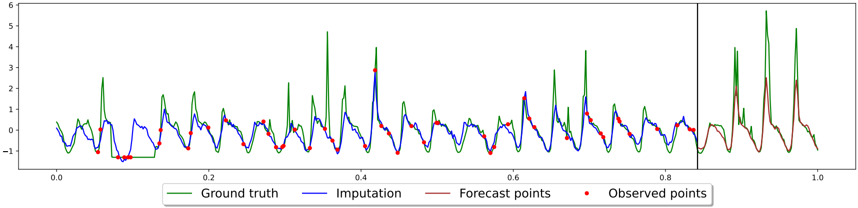



Figure: *Traffic dataset, sample 95*. In this figure, TimeFlow simultaneously imputes and forecasts at horizon 96 with a 10% partially observed look-back window of length 512.

# Quantify uncertainty with TimeFlow ($\mathcal{L}$ is the pinball loss)

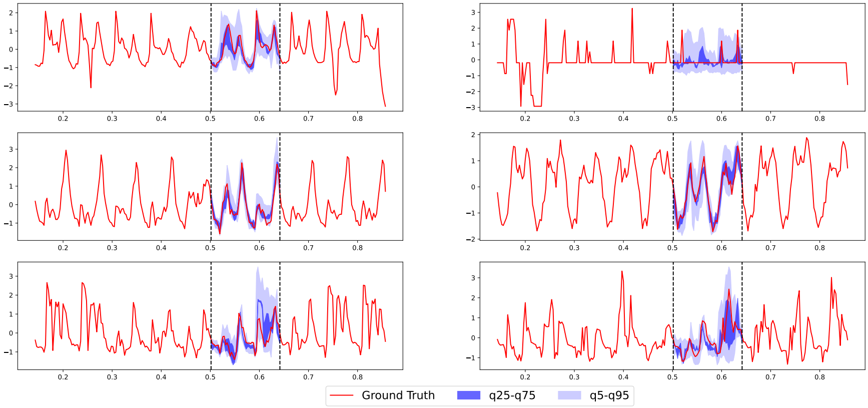



Figure: Quantifying uncertainty in block imputation of two missing days in the *Electricity* dataset.

# Conclusion

# Key takeaways

TimeFlow offers:

- A unified and continuous approach for time series imputation and forecasting.
- Adaptability to new contexts through meta-learning optimization.
- Extraction of semantically rich and practically useful representations for downstream tasks.

# A team work

**Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations**

**Etienne Le Naour[* 1,2], Louis Serrano[* 1], Léon Migus[* 1,3], Yuan Yin[1], Ghislain Agoua[2]**
**Nicolas Baskiotis[1], Patrick Gallinari[1,4], Vincent Guigue[5]**
[1] Sorbonne Université, CNRS, ISIR, 75005 Paris, France
[2] EDF R&D, Palaiseau, France
[3] Sorbonne Université, CNRS, Laboratoire Jacques-Louis Lions, 75005 Paris, France
[4] Criteo AI Lab, Paris, France
[5] AgroParisTech, Palaiseau, France
{*louis.serrano, leon.migus, yuan.yin, nicolas.baskiotis, vincent.guigue*}@*sorbonne-universite.fr*
{*etienne.le-naour, ghislain.agoua*}@*edf.fr*

- Click on this link

# References I

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.

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.

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.

# References II

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.

Y. Yin, M. Kirchmeyer, J.-Y. Franceschi, A. Rakotomamonjy, and P. Gallinari. Continuous pde dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*, 2022.

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.