AgroParisTech

AGRIAL

Sim-to-Real Transfer with Asymmetric Temporal Supervision

Using Alignment techniques on an Implicit Neural Representation (INR) based architecture

Julian Agudelo

Supervised by Cristina Manfredotti, Vincent Guigue, Evelyne Lutton and Hadrien Piot

MIA
PARIS-SACLAY
EKINOCS

23/04/2026

PLAN

1. A Reminder on Transfer Learning
2. Task Formulation
3. Lessons From Unsupervised Domain Alignment
4. Modulated Implicit Neural Representations
5. Domain Alignment on an INR-FiLM architecture
6. Use Case
7. Results

A Reminder on Transfer Learning

Domain and Task

An arbitrary **domain** is defined by a **feature space** $\mathcal{X}$ and a **marginal distribution** $P(X)$.

$$\mathcal{A} = \{\mathcal{X}, P(X)\}$$

An arbitrary **task** is defined by a **label space** $\mathcal{Y}$ and a **conditional distribution** $P(Y \mid X)$, induced by an unknown predictive function $f(X) = Y$

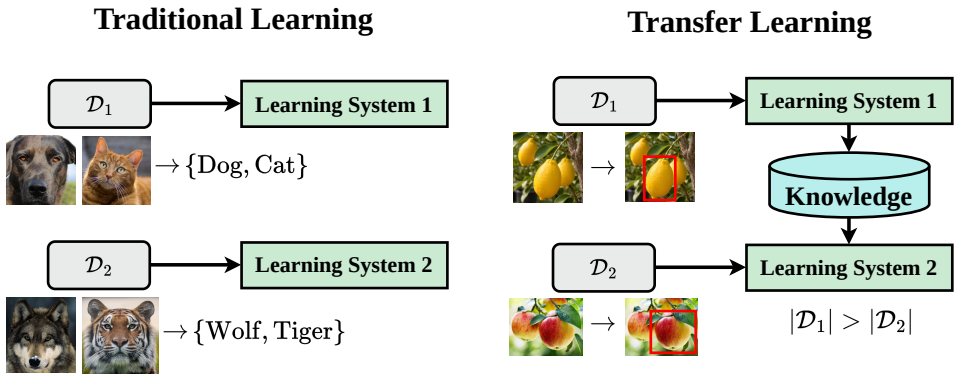
$$\mathcal{T}_{\mathcal{A}} = \{\mathcal{Y}, f\}$$

A domain and its associated task induce a dataset

$$\mathcal{D}_{\mathcal{A}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{N}$$

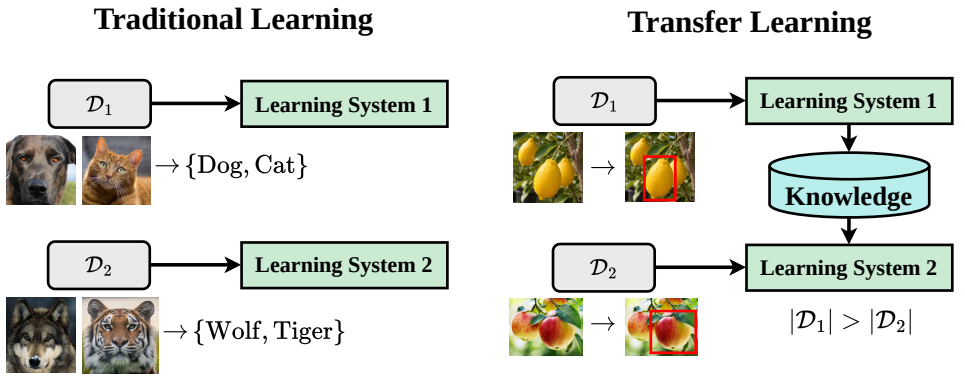
X and Y are random variables, so that $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$

The Classical Transfer Learning Framework



Pan and Yang 2010

The Classical Transfer Learning Framework



Covariate shift ($P_{\mathcal{S}}(X) \neq P_{\mathcal{R}}(X)$)

Label shift ($P_{\mathcal{S}}(Y) \neq P_{\mathcal{R}}(Y)$)

Concept shift ($P_{\mathcal{S}}(Y \mid X) \neq P_{\mathcal{R}}(Y \mid X)$)

Pan and Yang 2010

Foundations Models: Is **Transfer Learning** a **solved** problem?

What **model of the world** do they have?

Antoine Cornuéjols

AgroParisTech – INRAE MIA Paris-Saclay

EKINOCS research group

AgroParisTech

INRAE

The Classical Paradigm (Pre-2020)

- Neural networks learn **hierarchical representations**
- Each new domain or task requires retraining

Transfer Learning in the Era of Foundation Models

The Classical Paradigm (Pre-2020)

- Neural networks learn **hierarchical representations**
- Each new domain or task requires retraining

Foundation Models (Post-2020)

- FMs learn **universal representations** from massive unlabeled datasets
- Zero-shot capacities
- Each new domain requires **no retraining**, only lightweight adaptation

However...

- Many domains suffer from data scarcity: medical imaging, materials science, industrial fault diagnosis, ...
- Foundation Models learn statistical correlations at scale but have no mechanism to encode causal structure, physical laws, or hard constraints
- Adapation requires dense supervision in the target domain, making low-label regimes out of scope

Task Formulation

- We denote $X \in \mathcal{X}$ as $\mathbf{x} = \{\mathbf{x}^{(d)} \cup \mathbf{x}^{(s)}\}$
- The label $Y \in \mathcal{Y}$ is a time series $\{y_\tau\}_{\tau=0}^{T}$

- We denote $X \in \mathcal{X}$ as $\mathbf{x} = \{\mathbf{x}^{(d)} \cup \mathbf{x}^{(s)}\}$
- The label $Y \in \mathcal{Y}$ is a time series $\{y_\tau\}_{\tau=0}^{T}$

A key characteristic of our problem is that the two domains differ in the observations available for each label time series

- We denote $X \in \mathcal{X}$ as $\mathbf{x} = \{\mathbf{x}^{(d)} \cup \mathbf{x}^{(s)}\}$
- The label $Y \in \mathcal{Y}$ is a time series $\{y_\tau\}_{\tau=0}^{T}$

A key characteristic of our problem is that the two domains differ in the observations available for each label time series

In the simulated domain $\mathcal{S}$

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_{P_{\mathcal{S}}} \left[\sum_{t=0}^{T} \ell(\hat{y}_\tau, \, y_\tau) \right]$$

$$\mathcal{L}_{\mathcal{S}}(\theta) = \frac{1}{N_{\mathcal{S}}} \sum_{i=1}^{N_{\mathcal{S}}} \sum_{\tau=0}^{T} \ell\big(\hat{y}_\tau^{(i)}, \, y_\tau^{(i)}\big)$$

- We denote $X \in \mathcal{X}$ as $\mathbf{x} = \{\mathbf{x}^{(d)} \cup \mathbf{x}^{(s)}\}$
- The label $Y \in \mathcal{Y}$ is a time series $\{y_\tau\}_{\tau=0}^{T}$

A key characteristic of our problem is that the two domains differ in the observations available for each label time series

In the real-world domain $\mathcal{R}$

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_{P_{\mathcal{R}}} \left[\sum_{t=0}^{T} \ell(\hat{y}_\tau, \, y_\tau) \right]$$

$$\mathcal{L}_{\mathcal{R}}(\theta) = \frac{1}{N_{\mathcal{R}}} \sum_{j=1}^{N_{\mathcal{R}}} \left[\ell\big(\hat{y}_0^{(j)}, \, y_0^{(j)}\big) + \ell\big(\hat{y}_T^{(j)}, \, y_T^{(j)}\big) \right]$$

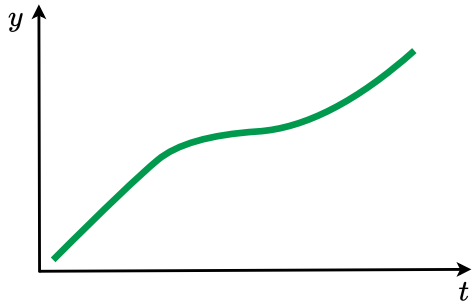
- We denote $X \in \mathcal{X}$ as $\mathbf{x} = \{\mathbf{x}^{(d)} \cup \mathbf{x}^{(s)}\}$
- The label $Y \in \mathcal{Y}$ is a time series $\{y_\tau\}_{\tau=0}^{T}$

A key characteristic of our problem is that the two domains differ in the observations available for each label time series

In the real-world domain $\mathcal{R}$

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_{P_{\mathcal{R}}} \left[\sum_{t=0}^{T} \ell(\hat{y}_\tau, \, y_\tau) \right]$$

$$\mathcal{L}_{\mathcal{R}}(\theta) = \frac{1}{N_{\mathcal{R}}} \sum_{j=1}^{N_{\mathcal{R}}} \left[\ell\big(\hat{y}_0^{(j)}, \, y_0^{(j)}\big) + \ell\big(\hat{y}_T^{(j)}, \, y_T^{(j)}\big) \right]$$

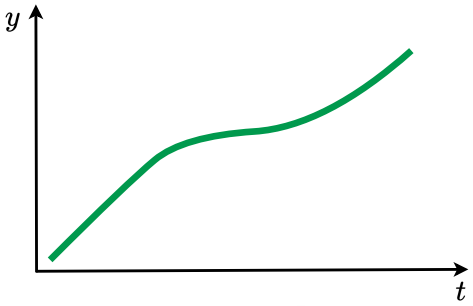


A sample from $D_{\mathcal{S}}$

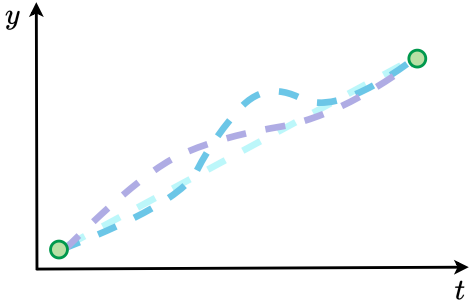
A sample from $D_{\mathcal{R}}$

Available y_t for supervision

Transfer Learning for Time Series with Asymmetric Supervision



A sample from $D_{\mathcal{S}}$

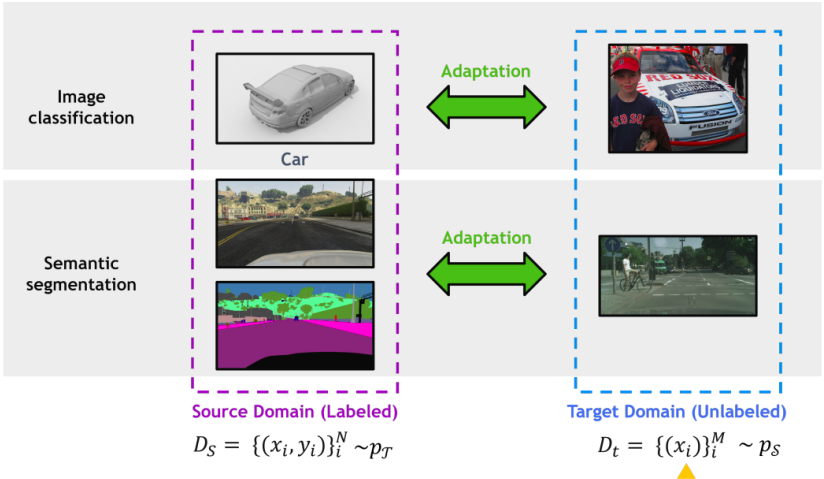


A sample from $D_{\mathcal{R}}$

Available y_t for supervision

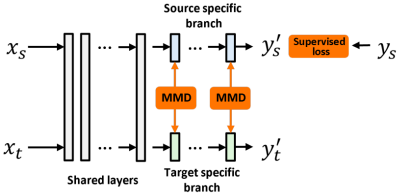
Lessons From Unsupervised Domain Alignment

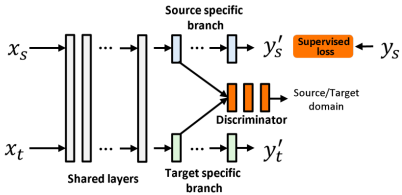
Unsupervised Domain Adaptation



Liu et al. 2022

Unsupervised Domain Adaptation





$$\mathcal{L}_{\text{MMD}} = \mathbb{E}_{x,x'}[k(x,x')] + \mathbb{E}_{y,y'}[k(y,y')] - 2\,\mathbb{E}_{x,y}[k(x,y)],$$

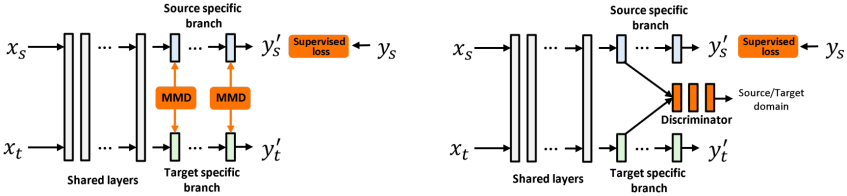
where $x, x' \sim z_{\text{synth}}$, $y, y' \sim z_{\text{real}}$, and the kernel function is $k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

$$\mathcal{L}_{\text{adv}} = -\sum_i \left[y_i \log \sigma(f_i) + (1-y_i) \log\left(1-\sigma(f_i)\right) \right],$$

where $y_i = 0$ denotes source samples, $y_i = 1$ denotes target samples, f_i is the discriminator logit, and $\sigma(\cdot)$ is the sigmoid function.

Liu et al. 2022

Unsupervised Domain Adaptation



$$\mathcal{L}_{\text{MMD}} = \mathbb{E}_{x,x'}[k(x,x')] + \mathbb{E}_{y,y'}[k(y,y')] - 2\,\mathbb{E}_{x,y}[k(x,y)],$$

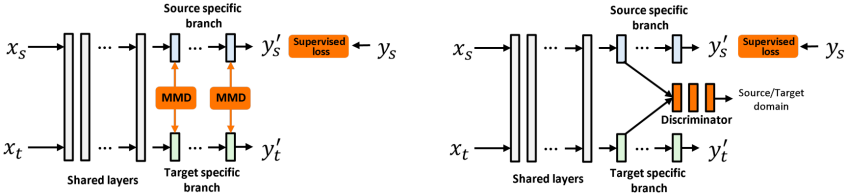
where $x, x' \sim z_{\text{synth}}$, $y, y' \sim z_{\text{real}}$, and the kernel function is $k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

$$\mathcal{L}_{\text{adv}} = -\sum_i \left[y_i \log \sigma(f_i) + (1-y_i)\log\left(1-\sigma(f_i)\right)\right],$$

where $y_i = 0$ denotes source samples, $y_i = 1$ denotes target samples, f_i is the discriminator logit, and $\sigma(\cdot)$ is the sigmoid function.

Liu et al. 2022

Unsupervised Domain Adaptation



$$\mathcal{L}_{\text{MMD}} = \mathbb{E}_{x,x'}[k(x,x')] + \mathbb{E}_{y,y'}[k(y,y')] - 2\,\mathbb{E}_{x,y}[k(x,y)],$$

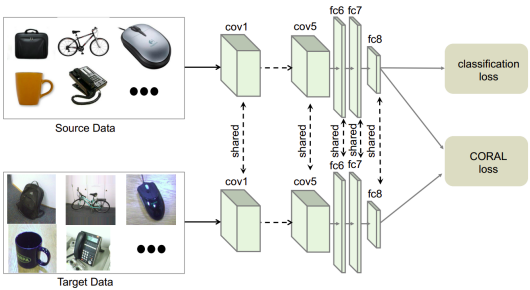
where $x, x' \sim z_{\text{synth}}$, $y, y' \sim z_{\text{real}}$, and the kernel function is $k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

$$\mathcal{L}_{\text{adv}} = -\sum_i \left[y_i \log \sigma(f_i) + (1-y_i) \log\left(1-\sigma(f_i)\right) \right],$$

where $y_i = 0$ denotes source samples, $y_i = 1$ denotes target samples, f_i is the discriminator logit, and $\sigma(\cdot)$ is the sigmoid function.

Liu et al. 2022

Unsupervised Domain Adaptation: CORAL



$$\mathcal{L}_{\text{CORAL}} = \frac{1}{4d^2} \|C_s - C_r\|_F^2$$

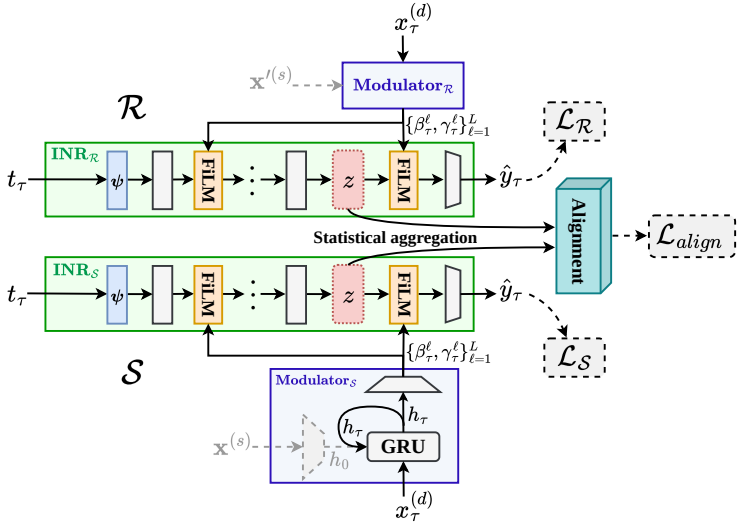
where $C_s, C_r \in \mathbb{R}^{d \times d}$ are the unbiased sample covariance matrices computed from the latent representations, and $\| \cdot \|_F$ denotes the Frobenius norm.

Sun and Saenko 2016

Unsupervised Domain Adaptation

- **Kernel-Based Distribution Matching (MMD):** aligns domain distributions in a Reproducing Kernel Hilbert Space (RKHS).
- **Adversarial Feature Alignment** aligns distributions implicitly via a minimax game between feature extractor and domain discriminator.
- **Alignment via Second-Order Statistics (CORAL)** Minimizes domain shift by matching covariance matrices.

The Proposed Architecture



Modulated Implicit Neural Representations

Implicit Neural Representations (INRs)

Implicit Neural Representations consider the problem of finding a **continuous representation of data**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Implicit Neural Representations (INRs)

Implicit Neural Representations consider the problem of finding a **continuous representation of data**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Multi Layer Perceptron **(MLPs)** $\rightarrow$ Continuous and fully differentiable.

INR

- Coordinate based MLPs
- Arbitrary resolution at inference
- Memory efficient

Explicit representations

- Point clouds, voxel grids, meshes, ...
- Resolution is fixed at acquisition time
- Less compact representations

Implicit Neural Representations (INRs)

Implicit Neural Representations consider the problem of finding a **continuous representation of data**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Multi Layer Perceptron **(MLPs)** $\rightarrow$ Continuous and fully differentiable.

INR

- Coordinate based MLPs
- Arbitrary resolution at inference
- Memory efficient

Explicit representations

- Point clouds, voxel grids, meshes, ...
- Resolution is fixed at acquisition time
- Less compact representations

A major problem is spectral Bias!

Implicit Neural Representations (INRs)

Implicit Neural Representations consider the problem of finding a **continuous representation of data**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Multi Layer Perceptron **(MLPs)** $\rightarrow$ Continuous and fully differentiable.

INR

- Coordinate based MLPs
- Arbitrary resolution at inference
- Memory efficient

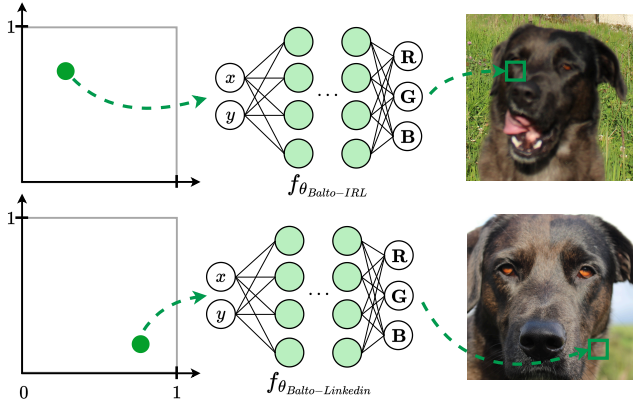
Explicit representations

- Point clouds, voxel grids, meshes, ...
- Resolution is fixed at acquisition time
- Less compact representations

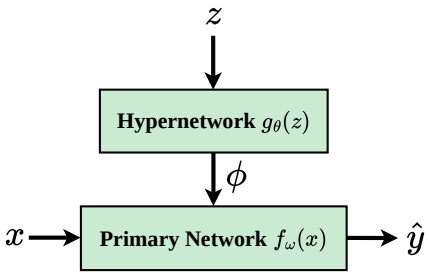
A major problem is spectral Bias!
Positional encoding or **adaptive activation functions** can capture a wide range of frequencies.

INRs for Learning Images Representations

Our function of interest could be simply an image, mapping the coordinates $[x, y]$ to their pixel values $[\mathbf{R}, \mathbf{G}, \mathbf{B}]$.

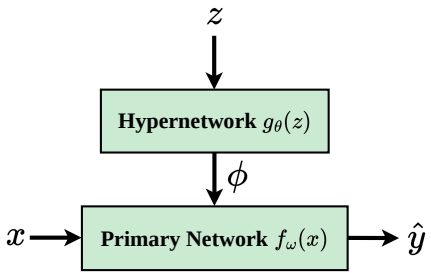


Hypernetworks and Feature-Wise Linear Modulation



$\phi \subseteq \omega$

Hypernetworks and Feature-Wise Linear Modulation



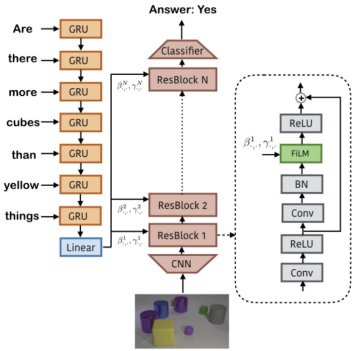
- ✔ A single hypernetwork can parameterize a family of primary networks
- ✔ High expressivity $\rightarrow$ Interesting for Multi-Task Learning
- ✖ Unstable and difficult to train
- ✖ Too expressive for most cases

Ha, Dai, and Le 2016

Hypernetworks and Feature-Wise Linear Modulation

Feature-wise Linear Modulation (FiLM) is a method for conditioning models by inserting feature-wise transformation layers into the primary network.

$$\mathrm{FiLM}(\mathbf{x}) = \gamma(z) \odot x + \beta(z)$$



- ✔ The FiLM generator (**modulator**) is an specialized Hypernetwork
- ✔ Sufficiently complex for most tasks
- ✔ Natural factorization of shared structure from instance-specific variation

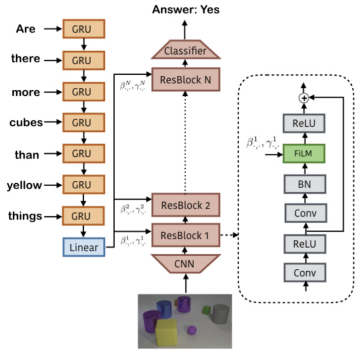
Useful for conditioning generative models, performing style transfer, . . .

Perez et al. 2017

Hypernetworks and Feature-Wise Linear Modulation

Feature-wise Linear Modulation (FiLM) is a method for conditioning models by inserting feature-wise transformation layers into the primary network.

$$\mathrm{FiLM}(\mathbf{x}) = \gamma(z) \odot x + \beta(z)$$

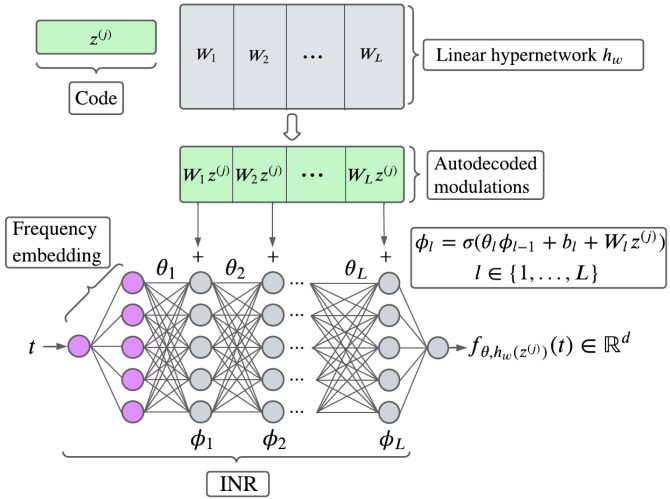


- ✔ The FiLM generator (**modulator**) is an specialized Hypernetwork
- ✔ Sufficiently complex for most tasks
- ✔ Natural factorization of shared structure from instance-specific variation **(nice for INRs ?)**

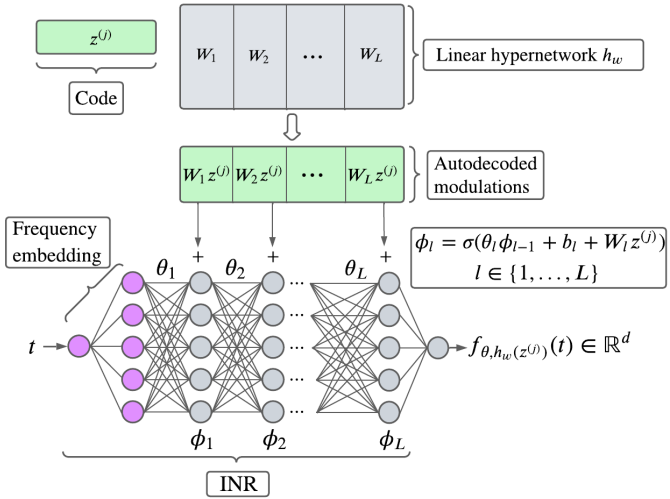
Useful for conditioning generative models, performing style transfer, . . .

Perez et al. 2017

Modulated INRs for Time Series Modeling

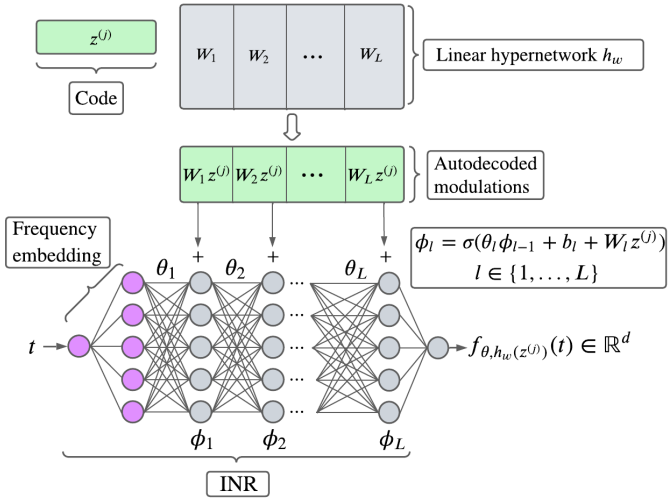


Modulated INRs for Time Series Modeling



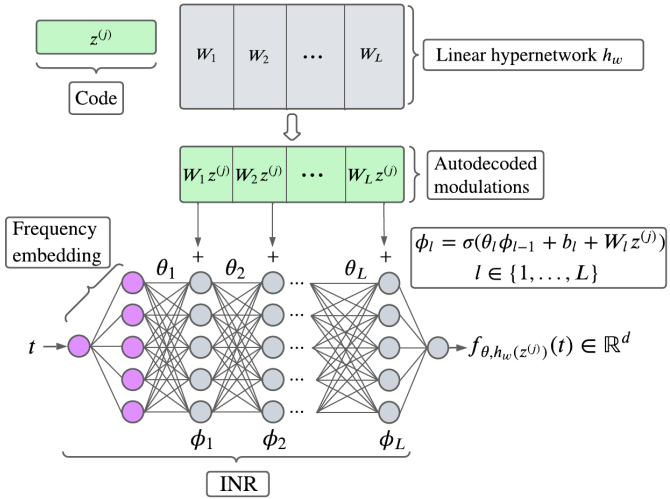
- ✔ SOTA performance on forecasting and imputation
- ✔ Test-Time Learning Capacities (3 SGD steps)

Modulated INRs for Time Series Modeling



- ✔ SOTA performance on forecasting and imputation
- ✔ Test-Time Learning Capacities (3 SGD steps)
- ✖ 2x slower at inference that other SOTA models
- ✖ Difficulty to fit time series with drastically different structure

Modulated INRs for Time Series Modeling



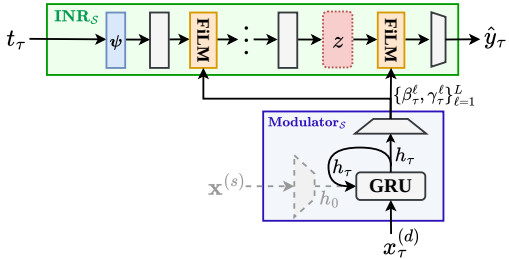
- ✔ SOTA performance on forecasting and imputation
- ✔ Test-Time Learning Capacities (3 SGD steps)
- ✖ 2x slower at inference that other SOTA models
- ✖ Difficulty to fit time series with drastically different structure

A beautiful example of using INRs for tasks other than representation

Naour et al. 2024

Domain Alignment on an INR-FiLM architecture

The Proposed Architecture



The Modulator receives a time series $\mathbf{x}^{(d)} = \{\mathbf{x}_\tau\}_{\tau=1}^T$ together with static covariates $\mathbf{x}^{(s)}$

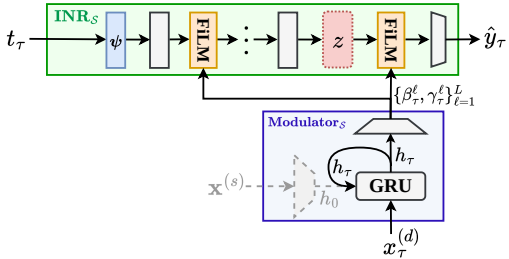
The static covariates initialise the hidden state of the GRU encoder through a MLP:

$$\mathbf{h}_0 = \mathrm{MLP}\left(\mathbf{x}^{(s)}\right)$$

Then, the GRU processes the dynamic inputs sequentially:

$$\mathbf{h}_\tau = \mathrm{GRU}\left(\mathbf{x}_\tau^{(d)}, \mathbf{h}_{\tau-1}\right), \qquad \tau = 1, \ldots, T$$

The Proposed Architecture



The Modulator receives a time series $\mathbf{x}^{(d)} = \{\mathbf{x}_\tau\}_{\tau=1}^T$ together with static covariates $\mathbf{x}^{(s)}$

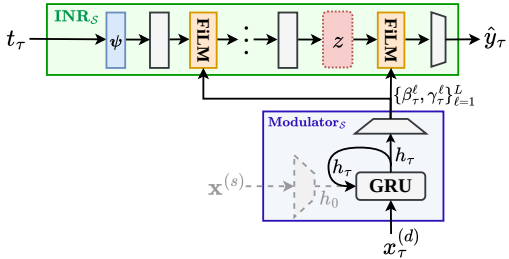
The static covariates initialise the hidden state of the GRU encoder through a MLP:

$$\mathbf{h}_0 = \mathrm{MLP}\left(\mathbf{x}^{(s)}\right)$$

Then, the GRU processes the dynamic inputs sequentially:

$$\mathbf{h}_\tau = \mathrm{GRU}\left(\mathbf{x}_\tau^{(d)}, \mathbf{h}_{\tau-1}\right), \qquad \tau = 1, \ldots, T$$

The Proposed Architecture



The Modulator receives a time series $\mathbf{x}^{(d)} = \{\mathbf{x}_\tau\}_{\tau=1}^T$ together with static covariates $\mathbf{x}^{(s)}$

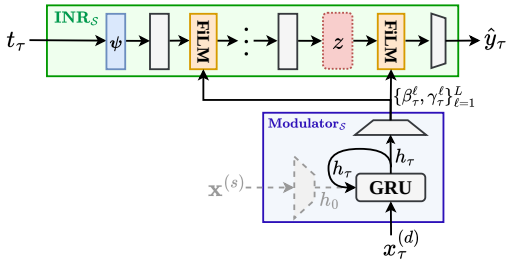
The static covariates initialise the hidden state of the GRU encoder through a MLP:

$$\mathbf{h}_0 = \mathrm{MLP}\left(\mathbf{x}^{(s)}\right)$$

Then, the GRU processes the dynamic inputs sequentially:

$$\mathbf{h}_\tau = \mathrm{GRU}\left(\mathbf{x}_\tau^{(d)}, \mathbf{h}_{\tau-1}\right), \qquad \tau = 1, \ldots, T$$

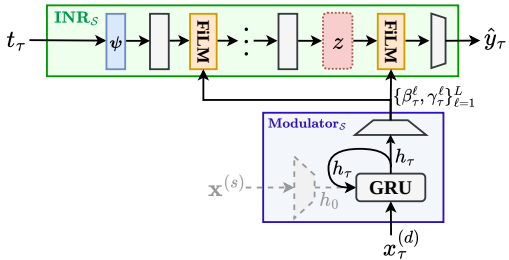
The Proposed Architecture



For each layer $\ell = 1, \ldots, L$ of the INR and for each step τ, the modulation parameters are obtained from the current GRU hidden state $\mathbf{h}_\tau$ through linear projections:

$$\boldsymbol{\gamma}_\tau^\ell = \mathbf{W}^{\ell,\gamma}\mathbf{h}_\tau + \mathbf{b}^{\ell,\gamma}, \qquad \boldsymbol{\gamma}_\tau^\ell, \boldsymbol{\beta}_\tau^\ell \in \mathbb{R}^d,$$
$$\boldsymbol{\beta}_\tau^\ell = \mathbf{W}^{\ell,\beta}\mathbf{h}_\tau + \mathbf{b}^{\ell,\beta},$$

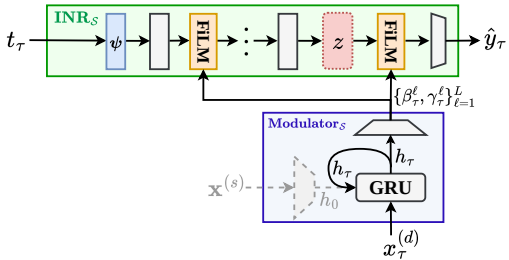
The Proposed Architecture



For each layer $\ell = 1, \ldots, L$ of the INR and for each step τ, the modulation parameters are obtained from the current GRU hidden state $\mathbf{h}_\tau$ through linear projections:

$$\boldsymbol{\gamma}_\tau^\ell = \mathbf{W}^{\ell,\gamma}\mathbf{h}_\tau + \mathbf{b}^{\ell,\gamma}, \qquad \boldsymbol{\gamma}_\tau^\ell, \boldsymbol{\beta}_\tau^\ell \in \mathbb{R}^d,$$
$$\boldsymbol{\beta}_\tau^\ell = \mathbf{W}^{\ell,\beta}\mathbf{h}_\tau + \mathbf{b}^{\ell,\beta},$$

The Proposed Architecture



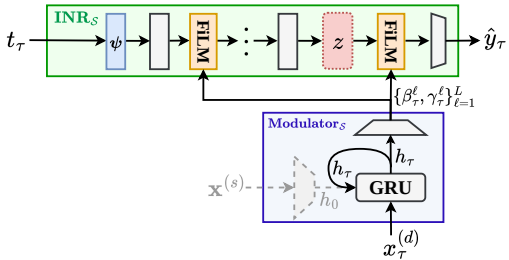
The INR is evaluated independently at each time step t_τ using modulation parameters generated from $\mathbf{h}_\tau$

First, we use a Fourier feature encoding $\psi : [0, 1] \to \mathbb{R}^{2K}$

$$\psi(t) = \left[\sin(2^0\pi t), \cos(2^0\pi t), \ldots, \sin(2^{K-1}\pi t), \cos(2^{K-1}\pi t)\right]$$

Mildenhall et al. 2020; Zhong et al. 2020

The Proposed Architecture



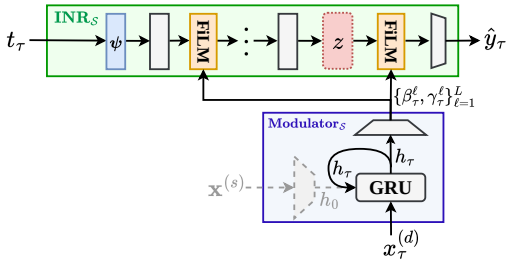
The INR is evaluated independently at each time step t_τ using modulation parameters generated from $\mathbf{h}_\tau$

First, we use a Fourier feature encoding $\boldsymbol{\psi} : [0, 1] \to \mathbb{R}^{2K}$

$$\boldsymbol{\psi}(t) = \left[\sin(2^0\pi t), \cos(2^0\pi t), \ldots, \sin(2^{K-1}\pi t), \cos(2^{K-1}\pi t)\right]$$

Mildenhall et al. 2020; Zhong et al. 2020

The Proposed Architecture



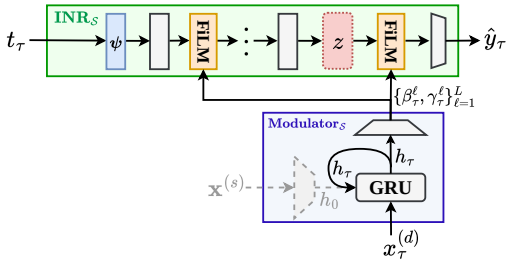
The INR is evaluated independently at each time step t_τ using modulation parameters generated from $\mathbf{h}_\tau$

First, we use a Fourier feature encoding $\boldsymbol{\psi} : [0, 1] \to \mathbb{R}^{2K}$

$$\boldsymbol{\psi}(t) = \left[\sin(2^0\pi t), \cos(2^0\pi t), \ldots, \sin(2^{K-1}\pi t), \cos(2^{K-1}\pi t)\right]$$

Mildenhall et al. 2020; Zhong et al. 2020

The Proposed Architecture



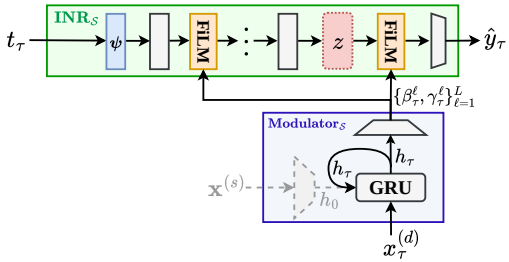
The first layer of **the INR** is $\mathbf{z}_\tau^{\ell=0} = \psi(t_\tau)$

Then, each of the remaining layers computes:

$$\mathbf{z}_\tau^\ell = \boldsymbol{\gamma}_\tau^\ell \odot \phi\big(\mathbf{W}^\ell \mathbf{z}_\tau^{\ell-1} + \mathbf{b}^\ell\big) + \boldsymbol{\beta}_\tau^\ell, \qquad \ell = 1, \ldots, L,$$

where ϕ denotes the activation function and $\odot$ the element-wise multiplication.

The Proposed Architecture



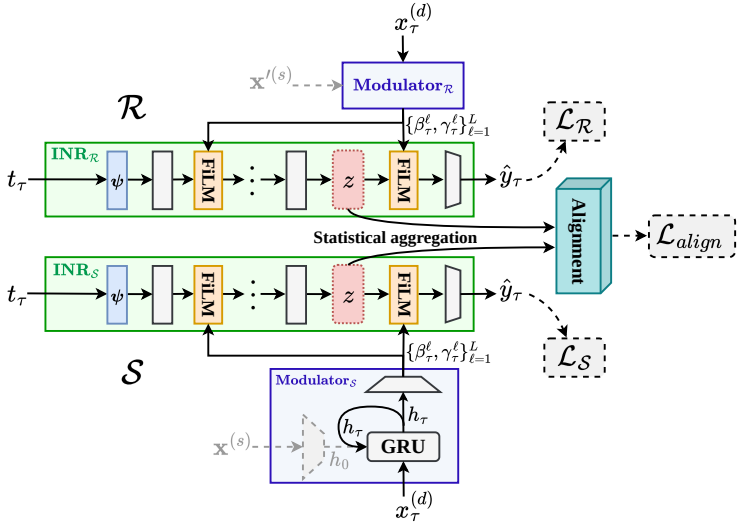
The first layer of **the INR** is $\mathbf{z}_\tau^{\ell=0} = \boldsymbol{\psi}(t_\tau)$

Then, each of the remaining layers computes:

$$\mathbf{z}_\tau^\ell = \boldsymbol{\gamma}_\tau^\ell \odot \phi\big(\mathbf{W}^\ell \mathbf{z}_\tau^{\ell-1} + \mathbf{b}^\ell\big) + \boldsymbol{\beta}_\tau^\ell, \qquad \ell = 1, \ldots, L,$$

where ϕ denotes the activation function and $\odot$ the element-wise multiplication.

The Proposed Architecture



Alignement Techniques

MMD

$$\mathcal{L}_{\text{MMD}} = \mathbb{E}_{x,x'}[k(x,x')] + \mathbb{E}_{y,y'}[k(y,y')] - 2\,\mathbb{E}_{x,y}[k(x,y)]$$

CORAL

$$\mathcal{L}_{\text{CORAL}} = \frac{1}{4d^2}\|C_s - C_r\|_F^2$$

Adv(μ, σ)

$$\mathcal{L}_{\text{adv}} = -\sum_i [y_i \log \sigma(f_i) + (1-y_i)\log(1-\sigma(f_i))]$$

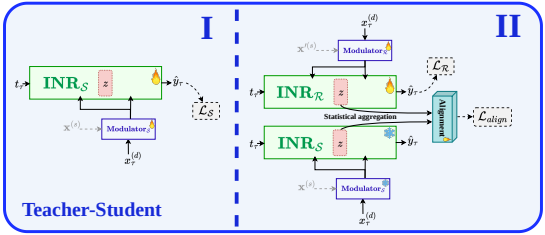
$$f_i = \text{MLP}([\mu_i, \sigma_i]), \quad \mu_i = \frac{1}{B}\sum_b h_b, \quad \sigma_i = \text{std}_b(h_b)$$

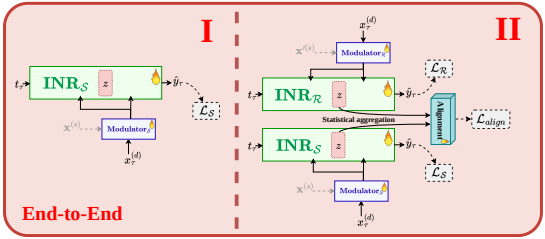
Adv(cov)

$$\mathcal{L}_{\text{adv}} = -\sum_i [y_i \log \sigma(f_i) + (1-y_i)\log(1-\sigma(f_i))]$$

$$f_i = \mathbf{w}^\top \frac{1}{H}\sum_{j=1}^{H} \text{MLP}(\mathbf{C}_i[j,:]), \quad \mathbf{C}_i = \frac{1}{B-1}\tilde{H}_i^\top \tilde{H}_i$$

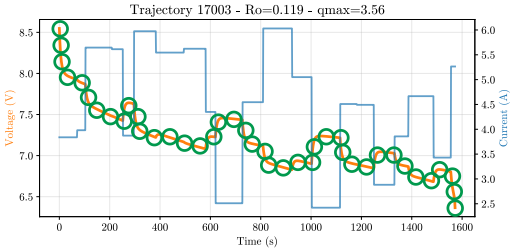
Training Setups



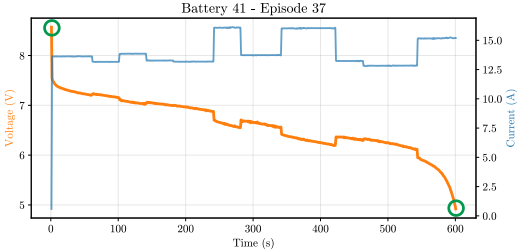


Use Case

Batteries Discharge Curve Prediction



A sample from $D_{\mathcal{S}}$



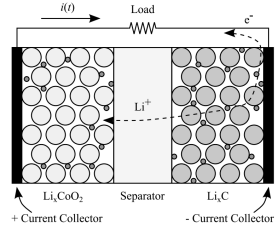
A sample from $D_{\mathcal{R}}$

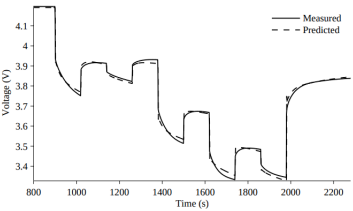
Available y_t for supervision

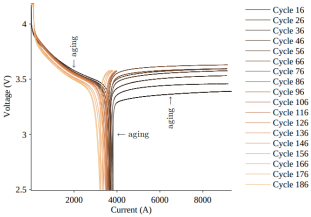
- 20000 samples: 1S, 2S and 3S
- R_0 and $q_{\max}$ (LHS)
- $\mathbf{x}^{(s)} = \{R_0, q_{\max}\}$
- 70/15/15, length $[100 - 3834]$

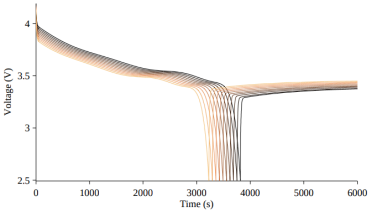
- 4382 samples from 15 batteries (2S)
- $\mathbf{x}^{(s)} = \{\text{Cycle\_number}\}$
- 70/15/15, length $[152 - 3693]$

Electrochemical Simulator



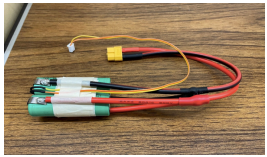


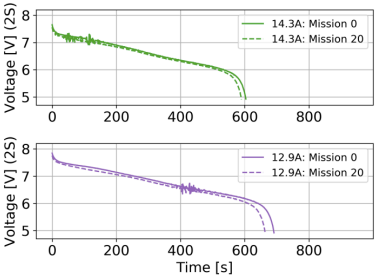




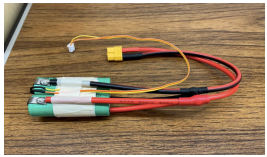
Daigle and Kulkarni 2013

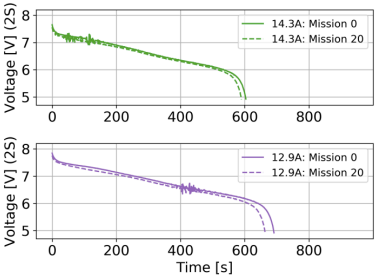
Samsung INR18650-25R





Samsung INR18650-25R





- The batteries are the same model as the ones used for the physical model's default parameters !
- We know the entire voltage sequence
- **Non-uniform time step** and **good representation of sensor artifacts**.

Fricke et al. 2023

Results

Quantitative Results

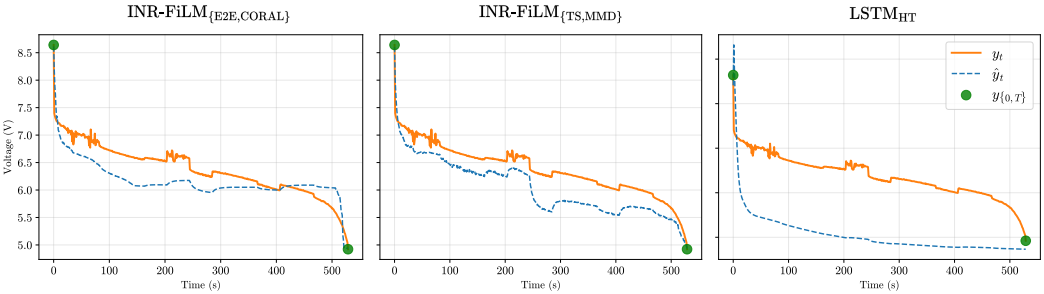
Alignment method	Simulated Domain $\mathcal{S}$				Real-world Domain $\mathcal{R}$					
	T-S		E2E		T-S			E2E		
	MSE	$\mathbf{V}_{mon}$	MSE	$\mathbf{V}_{mon}$	$\text{MSE}_{\{0,T\}}$	$\mathbf{V}_{mon}$	MSE	$\text{MSE}_{\{0,T\}}$	$\mathbf{V}_{mon}$	MSE
MMD	0.075	0.397	0.084	0.369	9.41e-4	0.380	**0.337**	9.29e-4	0.363	0.301
CORAL	0.074	0.377	0.081	0.397	**7.18e-4**	0.062	0.494	**7.84e-4**	**0.199**	**0.147**
Adv(μ, σ)	0.071	0.333	0.091	0.425	5.65e-4	0.451	0.391	1.98e-3	0.475	0.870
Adv(cov)	0.075	0.384	0.086	0.394	8.26e-4	**0.028**	0.897	1.43e-3	0.407	0.446

Quantitative Results

Alignment method	Simulated Domain $\mathcal{S}$				Real-world Domain $\mathcal{R}$					
	T-S		E2E		T-S			E2E		
	MSE	$\mathbf{V}_{mon}$	MSE	$\mathbf{V}_{mon}$	$\text{MSE}_{\{0,T\}}$	$\mathbf{V}_{mon}$	MSE	$\text{MSE}_{\{0,T\}}$	$\mathbf{V}_{mon}$	MSE
MMD	0.075	0.397	0.084	0.369	9.41e-4	0.380	**0.337**	9.29e-4	0.363	0.301
CORAL	0.074	0.377	0.081	0.397	**7.18e-4**	0.062	0.494	**7.84e-4**	**0.199**	**0.147**
Adv(μ, σ)	0.071	0.333	0.091	0.425	5.65e-4	0.451	0.391	1.98e-3	0.475	0.870
Adv(cov)	0.075	0.384	0.086	0.394	8.26e-4	**0.028**	0.897	1.43e-3	0.407	0.446

Model	$\text{MSE}_{\{0,T\}}$	MSE
LSTM_{FT}	0.371	1.851
LSTM_{HT}	0.434	1.487
INR_{FT}	0.366	2.968
INR_{mod}	0.363	2.600
$\text{INR} - \text{FiLM}_{\{\text{E2E,CORAL}\}}$ **(Best Ours)**	**7.84e-4**	**0.147**

Qualitative Results

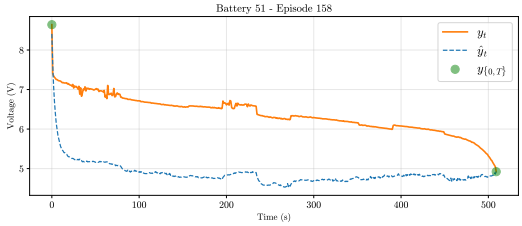


So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**



- ✖ Lack of quantitative evaluation

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**
- ✔ Continuous representations are interesting in Sim-to-Real Transfer

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**
- ✔ Continuous representations are interesting in Sim-to-Real Transfer
- ✔ Our main contribution are open questions:

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**
- ✔ Continuous representations are interesting in Sim-to-Real Transfer
- ✔ Our main contribution are open questions:
  - How much Expert Knowledge is enough?

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**
- ✔ Continuous representations are interesting in Sim-to-Real Transfer
- ✔ Our main contribution are open questions:
  - How much Expert Knowledge is enough?
  - How can we build metrics?

So What?

- ✖ Sensitivity to simulation coverage $\rightarrow$ **Expert Knowledge**
- ✖ Lack of quantitative evaluation
- ✔ **The results are promising!** we addressed a very difficult inverse problem **compensating the lack of supervision with the knowledge gained from learning a surrogate**
- ✔ Continuous representations are interesting in Sim-to-Real Transfer
- ✔ Our main contribution are open questions:
  - How much Expert Knowledge is enough?
  - How can we build metrics?
  - Is our task feasible?

MERCI


WHEN IT'S BETTER TO CALL YOURSELF A...
"STUDENT"
"RESEARCHER"
GETTING A STUDENT DISCOUNT
REGISTERING FOR A CONFERENCE
USING STUDENT FACILITIES (HEALTH CENTER, GYM, HOUSING)
TALKING TO FRIENDS AND FAMILY
CRASHING DEPARTMENT PARTIES WITH SIGNS THAT SAY "NO STUDENTS ALLOWED"
FINDING PARKING ON CAMPUS
IMPRESSING SOMEONE ON A FIRST DATE
JORGE CHAM © 2016
WWW.PHDCOMICS.COM

References I

- Daigle, Matthew and Chetan S Kulkarni (2013). “Electrochemistry-based Battery Modeling for Prognostics”. en. In.
- Fricke, Kajetan et al. (2023). “An Accelerated Life Testing Data set for Lithium-Ion Batteries with Constant and Variable Loading Conditions”. en. In.
- Ha, David, Andrew Dai, and Quoc V. Le (2016). *HyperNetworks*. arXiv: 1609.09106 [cs.LG]. URL: https://arxiv.org/abs/1609.09106.
- Jin, Xiaoyong et al. (2022). *Domain Adaptation for Time Series Forecasting via Attention Sharing*. arXiv: 2102.06828 [cs.LG]. URL: https://arxiv.org/abs/2102.06828.
- Liu, Xiaofeng et al. (2022). *Deep Unsupervised Domain Adaptation: A Review of Recent Advances and Perspectives*. arXiv: 2208.07422 [cs.CV]. URL: https://arxiv.org/abs/2208.07422.

References II

- Mildenhall, Ben et al. (Aug. 2020). *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. en. arXiv:2003.08934 [cs]. DOI: 10.48550/arXiv.2003.08934. URL: http://arxiv.org/abs/2003.08934 (visited on 03/25/2026).
- Naour, Etienne Le et al. (2024). *Time Series Continuous Modeling for Imputation and Forecasting with Implicit Neural Representations*. arXiv: 2306.05880 [cs.LG]. URL: https://arxiv.org/abs/2306.05880.
- Ozyurt, Yilmazcan, Stefan Feuerriegel, and Ce Zhang (2023). "Contrastive Learning for Unsupervised Domain Adaptation of Time Series". In: *ICLR*.
- Pan, Sinno Jialin and Qiang Yang (Oct. 2010). "A Survey on Transfer Learning". en. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191. URL: http://ieeexplore.ieee.org/document/5288526/ (visited on 03/23/2026).
- Perez, Ethan et al. (2017). *FiLM: Visual Reasoning with a General Conditioning Layer*. arXiv: 1709.07871 [cs.CV]. URL: https://arxiv.org/abs/1709.07871.

References III

Sun, Baochen and Kate Saenko (2016). *Deep CORAL: Correlation Alignment for Deep Domain Adaptation*. arXiv: 1607.01719 [cs.CV]. URL: https://arxiv.org/abs/1607.01719.

Zhong, Ellen D. et al. (Feb. 2020). *Reconstructing continuous distributions of 3D protein structure from cryo-EM images*. en. arXiv:1909.05215 [q-bio]. DOI: 10.48550/arXiv.1909.05215. URL: http://arxiv.org/abs/1909.05215 (visited on 03/25/2026).