# Adapting Foundation Models: From Reinforcement Learning to Multivariate Time Series Forecasting

Abdelhakim Benechehab[1,2]

[1]Huawei Paris Noah's Ark Lab
[2]Department of Data Science, EURECOM

abdelhakim.benechehab@gmail.com

-

December 11, 2025

# About Me

- **2016–2020** — MEng, *École des Mines*
  - Major: Data Science

- **2020–2021** — MSc, *ENS Paris-Saclay*
  - Applied Mathematics (MVA Program)
  - Research internship: *InstaDeep* (Model-based RL)

- **2021–2023** — Research Engineer
  - Huawei Noah's Ark Lab (Model-based RL)

- **2023–2026** — PhD, *EURECOM (Sorbonne University) & Huawei Noah's Ark Lab*
  - Thesis topic: Model-based Reinforcement Learning in the era of Foundation Models

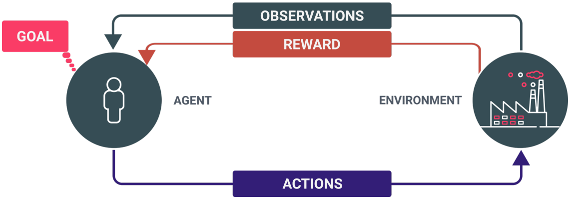
# Outline

# Preliminaries

# Reinforcement Learning

Reinforcement Learning environments are Markov decision processes $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \mu_0, \gamma \rangle$, where:

- $S$ state space, $A$ action space.
- Transition fn $P_t : (s, a, s') \mapsto \mathbf{Pr}(s_{t+1} = s' | s_t = s, a_t = a)$.
- Reward function $r : (s, a) \mapsto r(s, a)$.
- $\mu_0$ initial state distribution, $\gamma \in [0, 1]$ discount factor.



# Reinforcement Learning

The goal of RL is to find a policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ that maximizes the return:

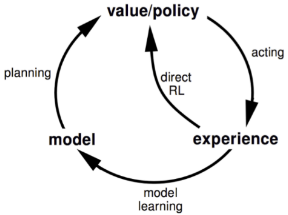
$$\eta(\pi) := \mathbb{E}_{s_0 \sim \mu_0, a_t \sim \pi,\, s_{t>0} \sim P_t}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

# Model-based Reinforcement Learning

**Model-based RL (MBRL)** learns the transition $\hat{P}$ from interaction data. The model maximizes the log-likelihood:

$$\mathcal{L}(\mathcal{D}; \hat{P}) = \frac{1}{N} \sum_{i=1}^{N} \log \hat{P}(s_{t+1}^i | s_t^i, a_t^i)$$

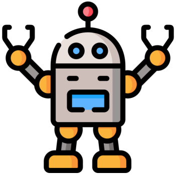
The learned model is used for policy search under the *learned* MDP $\widehat{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \hat{P}, r, \mu_0, \gamma \rangle$.

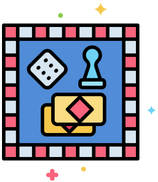


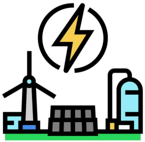

*towards data science - https://tinyurl.com/3kkxfzp4*

# Motivation

MBRL is particularly useful under **budget** and **safety** constraints.

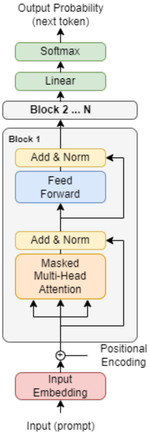


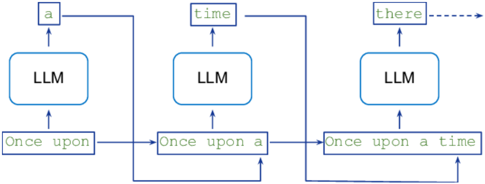




# Large Language Models (LLMs)

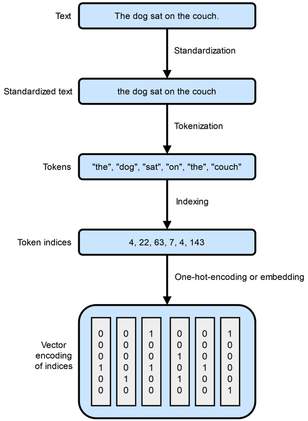
Large Language Models (LLMs) are **transformer**-based, **decoder-only** models trained using **autoregressive** next token prediction.



# LLMs: Text tokenization

Mapping words to vectors.



# LLMs: Numerical data tokenization

LLaMA 3 Tokenizer

- Digits: ['0', '1', ... '999']
- Token Ids: [15, 16, 17, ... 5500]

"151,167,...,267"

LLMTime

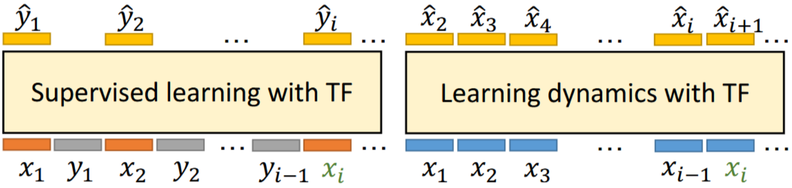
Time series processing

- Time series: [0.2513, 5.2387, 9.7889]
- Rescale+Encode: [150, 516, 850]
- Text: '150,516,850,'
- Tokens: ['150', ',', '516', ',', '850', ',']
- Token indices: [3965, 11, 20571, 11, 16217, 11]

Sampling: *Softmax* over the digits tokens' logits

# In-context Learning (ICL)

| In-context learning | Input prompt | Desired Output |
|---|---|---|
| Natural language processing | berry, baya, apple, manzana, banana | plátano |
| | Japan, mochi, France, croissant, Greece | baklava |
| Supervised learning $y_i = f(x_i) + \text{noise}$ | $x_1, y_1, x_2, \ldots, x_{i-1}, y_{i-1}, x_i$ | $f(x_i)$ |
| Dynamical systems $x_{i+1} = f(x_i) + \text{noise}$ | $x_1, x_2, x_3, \ldots, x_{i-2}, x_{i-1}, x_i$ | $f(x_i)$ |




*Transformers as Algorithms: Generalization and Stability in In-context Learning*

# Problem setup

# Dynamics learning using LLMs

- State space $\mathbb{R}^{d_s}$, Action space $\mathbb{R}^{d_a}$, Reward $\mathbb{R}$
- Given a trajectory

$$\tau^\pi = (s_0, a_0, s_1, a_1, s_2, \dots, s_{T-1})$$

We want to learn the distribution of the next state using ICL and a pre-trained LLM with parameters $\theta$:

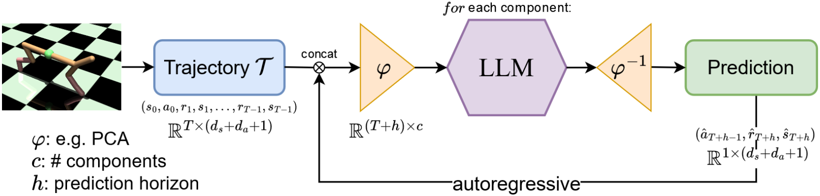
$$\{\hat{P}_\theta^{\pi,j}(s_t^j|\tau^\pi)\}_{t\leq T, j\leq d_s} = \mathsf{ICL}_\theta(\tau^\pi)$$

Challenges:

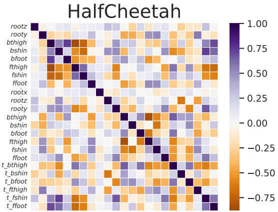
1. Multivariate states: $d_s > 1$
2. Including actions in-context: $P(s_t^j|s_0, a_0, s_1, a_1, s_2, \dots, s_{T-1})$

# Approach

**DICL**: Disentangled In-Context Learning [1]

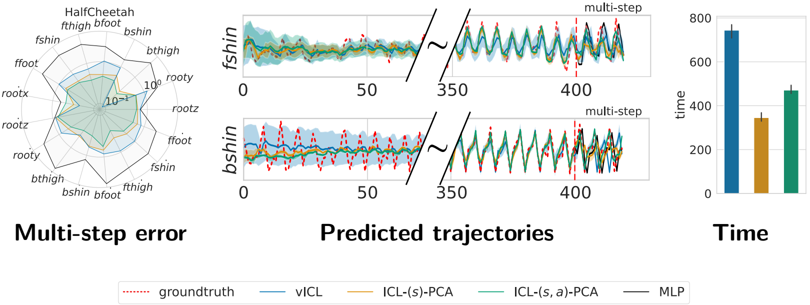



In practice, we project states and actions $(s, a)$ into the space of `PCA` components.



# Results

# Results: Prediction Error




**Multi-step error** **Predicted trajectories** **Time**

**PCA-based DICL achieves smaller multi-step error in less computational time.** We compare DICL-$(s)$ and DICL-$(s, a)$ using a number of components equal to half the number of features, with the vanilla approach vICL and an MLP baseline.

# Results: DICL-SAC

SAC: Soft Actor-Critic (an off-shelf RL algorithm)

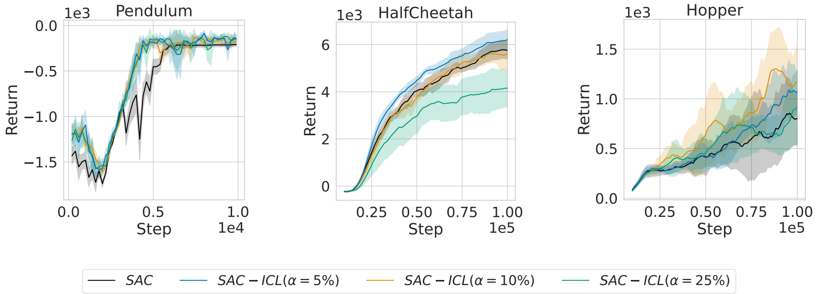
$+$

DICL

$=$

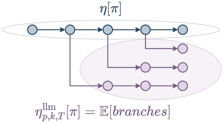
**DICL-SAC**

**for** $t = 1, \ldots, N\_interactions$ **do**
  New transition $(s_t, a_t, r_t, s_{t+1})$ from $\pi_\theta$
  Add $(s_t, a_t, r_t, s_{t+1})$ to $\mathcal{R}$
  Store auxiliary action $\tilde{a}_t \sim \pi_\theta(.|s_t)$
  **if** Generate LLM data **then**
    Sample trajectory $\mathcal{T} = (s_0, \ldots, s_{T_{\max}})$ from $\mathcal{R}$
    $\{\hat{s}_{i+1}\}_{0 \leq i \leq T_{\max}} \sim$ DICL-$(s)$ $(\mathcal{T})$
    Add $\{(s_i, \tilde{a}_i, r_i, \hat{s}_{i+1})\}_{T \leq i \leq T_{\max}}$ to $\mathcal{R}_{\text{llm}}$
  **end if**
  **if** update SAC **then**
    Sample batch $\mathcal{B}$ of size $b$ from $\mathcal{R}$
    Sample batch $\mathcal{B}_{\text{llm}}$ of size $\alpha \cdot b$ from $\mathcal{R}_{\text{llm}}$
    Update $\phi$ and $\psi$ on $\mathcal{B} \cup \mathcal{B}_{\text{llm}}$
  **end if**
**end for**



# Results: DICL-SAC (Theoretical guarantee)

Under mild assumptions on the LLM prediction error $\varepsilon_{\text{llm}}$, we have:




**Theorem (Multi-branch return bound)**

- $T$ *the context length*
- $p \in [0, 1]$ *probability of branching*
- $k$ *the branch length*
- $\varepsilon_{llm}$ *the LLM in-context learning prediction error*

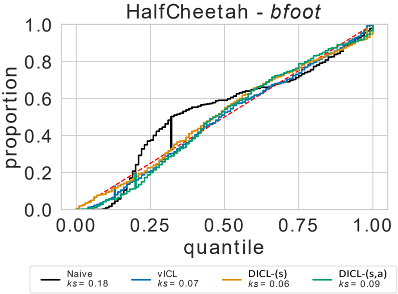
$$|\eta(\pi) - \eta^{llm}_{p,k,T}(\pi)| \leq 2\frac{\gamma^T}{1-\gamma} r_{\max} k^2 \, p \, \varepsilon_{llm}(T)$$

*where* $r_{\max} = \max_{s \in \mathcal{S}, a \in \mathcal{A}} r(s, a)$.

# Results: Calibration

**Quantile calibration:** For probabilistic regression, a perfectly calibrated forecaster means that $p\%$ of groundtruth values fall within the $p\%$-confidence interval of the predicted CDF.

LLMs are well-calibrated in-context forecasters.



# Outline

# Problem setup

# Problem setup

Consider a multivariate time series forecasting task:

- $\mathbf{X} \in \mathbb{R}^{L \times D}$ data matrix
- $\mathbf{Y} \in \mathbb{R}^{H \times D}$ target
  - $L$ lookback window (context length)
  - $H$ forecasting horizon
  - $D$ dimension (number of covariates)

We want to find the best adapter $\varphi^*$ such that:

**Definition (adapter)**

Feature-space transformation $\varphi : \mathbb{R}^D \to \mathbb{R}^{D'}$ such that:

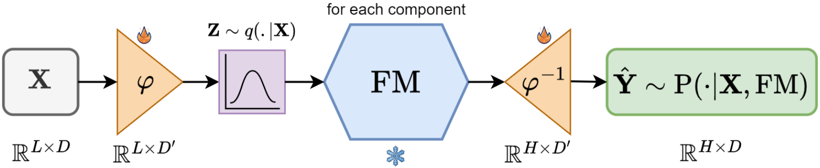
$$\hat{\mathbf{Y}}(\mathbf{X}; \varphi) = \varphi^{-1}\big(\texttt{FM}(\varphi(\mathbf{X}))\big), \text{ and } \varphi^* = \text{argmin}_{\varphi} \|\mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{X}; \varphi)\|_{\mathrm{F}}^2,$$

where FM is a fixed time series foundation model.

# Approach

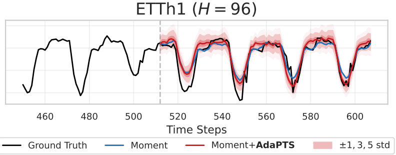
# AdaPTS

**AdaPTS**: Adapters for Probabilistic multivariate Time Series forcasting [2]




Properties:

1. Mixing features
2. Probabilistic predictions



# Theoretical analysis: Linear case

For a linear adapter $\varphi(\mathbf{X}) = \mathbf{X}\mathbf{W}_\varphi$ and a linear FM $f_{\mathrm{FM}}(\mathbf{X}) = \mathbf{W}_{\mathsf{FM}}^\top \mathbf{X} + \mathbf{b}_{\mathsf{FM}} \mathbf{1}^\top$, we have:

**Proposition (Optimal linear adapter)**

*The closed-form solution of the problem*

$$\min_{\mathbf{W}_\varphi} \mathcal{L}(\mathbf{W}_\varphi) = \|\mathbf{Y} - \left(\mathbf{W}_{FM}^\top \mathbf{X}\mathbf{W}_\varphi + \mathbf{b}_{FM}\mathbf{1}^\top\right)\mathbf{W}_\varphi^{-1}\|_F^2$$

*writes as:*

$$\mathbf{W}_\varphi^* = (\mathbf{B}^\top \mathbf{A})^+ \mathbf{B}^\top \mathbf{B},$$

*where* $\mathbf{A} = \mathbf{Y} - \mathbf{W}_{FM}^\top \mathbf{X}$ *and* $\mathbf{B} = \mathbf{b}_{FM}\mathbf{1}^\top$*.*

$\rightarrow$ Takeaway: The optimal solution is **not** the identity.

# Results

# Results: Forecasting error (MSE)

Families of adapters:

1. deterministic
   - Linear AutoEncoder
   - Deep non-linear AutoEncoder
   - Normalizing Flow
2. probabilistic
   - + Variational Inference
   - + MC Dropout

| Dataset | H | No adpt | with adapter | | | | |
|---|---|---|---|---|---|---|---|
| | | `Moment` | `PCA` | `LinAE` | `dropLAE` | `LinVAE` | `VAE` |
| ETTh1 | 96 | $0.411_{\pm.012}$ | $0.433_{\pm.001}$ | $0.402_{\pm.002}$ | $\mathbf{0.395_{\pm.003}}$ | $0.400_{\pm.001}$ | $0.404_{\pm.001}$ |
| | 192 | $\mathbf{0.431_{\pm.001}}$ | $0.440_{\pm.000}$ | $0.452_{\pm.002}$ | $0.446_{\pm.001}$ | $0.448_{\pm.002}$ | $\mathbf{0.431_{\pm.001}}$ |
| Ill | 24 | $2.902_{\pm.023}$ | $2.98_{\pm.001}$ | $2.624_{\pm.035}$ | $2.76_{\pm.061}$ | $2.542_{\pm.036}$ | $\mathbf{2.461_{\pm.008}}$ |
| | 60 | $3.000_{\pm.004}$ | $3.079_{\pm.000}$ | $3.110_{\pm.127}$ | $2.794_{\pm.015}$ | $\mathbf{2.752_{\pm.040}}$ | $2.960_{\pm.092}$ |
| Wth | 96 | $0.177_{\pm.010}$ | $0.176_{\pm.000}$ | $0.169_{\pm.000}$ | $\mathbf{0.156_{\pm.001}}$ | $0.161_{\pm.001}$ | $0.187_{\pm.001}$ |
| | 192 | $0.202_{\pm.000}$ | $0.208_{\pm.001}$ | $\mathbf{0.198_{\pm.001}}$ | $0.200_{\pm.001}$ | $0.204_{\pm.000}$ | $0.226_{\pm.000}$ |
| ExR | 96 | $\mathbf{0.130_{\pm.011}}$ | $0.147_{\pm.000}$ | $0.167_{\pm.013}$ | $\mathbf{0.130_{\pm.011}}$ | $0.243_{\pm.039}$ | $0.455_{\pm.010}$ |
| | 192 | $\mathbf{0.210_{\pm.002}}$ | $0.222_{\pm.000}$ | $0.304_{\pm.005}$ | $0.305_{\pm.013}$ | $0.457_{\pm.020}$ | $0.607_{\pm.021}$ |

# Results: Interpretable latent representations

Desirable representation learning properties:

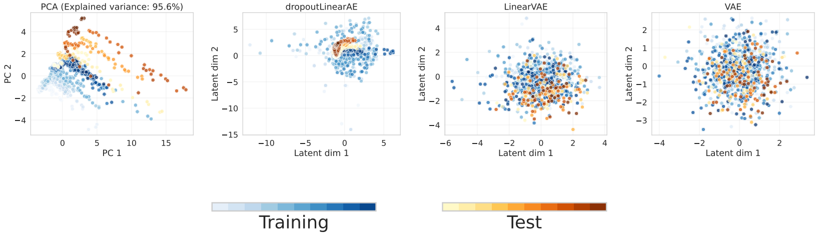
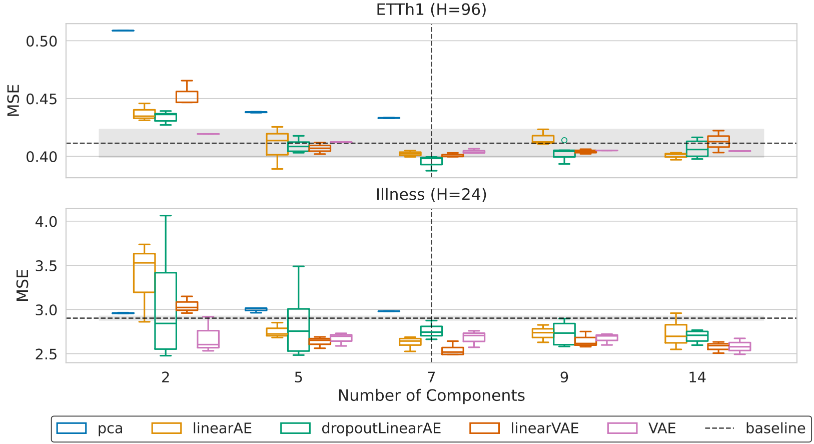



Figure: Visualization of the latent representation obtained by different adapters on Illness($H = 24$). Shaded colors indicate the time dimension, with lighter colors representing earlier timesteps.

# Results: Dimensionality reduction

Better forecasting accuracy even with lower dimensions



# Outline

# Conclusion

- We presented **DICL**, a methodology to adapt LLMs for the task of dynamics learning in MBRL
- We then presented **AdaPTS** a learning-based and probabilistic extension of adapters to multivariate time series forecasting

**Take Home Message**

**Foundation Models** are powerful predictors trained on vast amounts of data
$\rightarrow$ **Adapters** are an effective way to adapt FMs to custom problems

# References


A. Benechehab, Y. A. E. Hili, A. Odonnat, O. Zekri, A. Thomas, G. Paolo, M. Filippone, I. Redko, and B. Kégl, "Zero-shot model-based reinforcement learning using large language models," in *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

A. Benechehab, V. Feofanov, G. Paolo, A. Thomas, M. Filippone, and B. Kégl, "Adapts: Adapting univariate foundation models to probabilistic multivariate time series forecasting," *Forty-second International Conference on Machine Learning (ICML)*, May 2025.

# Thank You!

Want to know more?

Contact:

https://abenechehab.github.io/

abdelhakim.benechehab@gmail.com

@abenechehab

Slides available at:

https://abenechehab.github.io/assets/pdf/research.pdf