Zero-shot Model-based Reinforcement Learning using Large Language Models

Abdelhakim Benechehab[1,2], Youssef Attia El Hili[1], Ambroise Odonnat[1,3], Oussama Zekri[4], Albert Thomas[1], Giuseppe Paolo[1], Maurizio Filippone[5], Ievgen Redko[1], Balázs Kégl[1]

[1] Huawei Noah's Ark Lab, Paris, France
[2] Department of Data Science, EURECOM
[3] Inria, Univ. Rennes 2, CNRS, IRISA
[4] ENS Paris-Saclay
[5] Statistics Program, KAUST

–

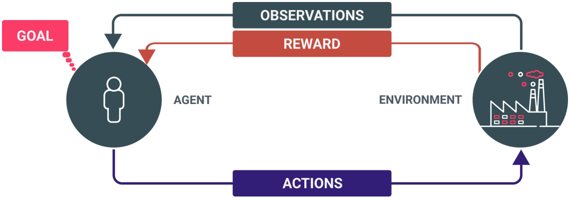
March 30, 2025

Outline

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 $r : (s, a) \mapsto r(s, a)$.
- μ_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:

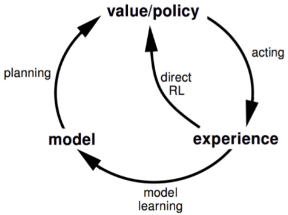
$$\eta(\pi) := \mathbb{E}_{s_0 \sim \mu_0, a_t \sim \pi, \, s_{t>0} \sim P_t}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

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

LLMs: Numerical data tokenization

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

LLaMA 3 Tokenizer

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

LLMTime

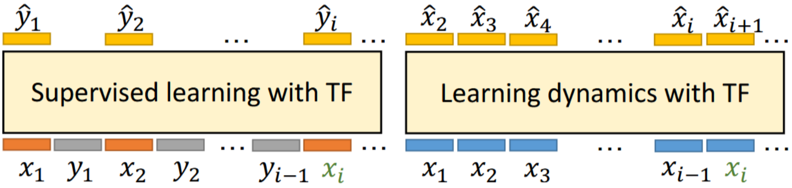
Time series processing

- Time series: [0.2513, 5.2387, 9.7889]
- Rescale+Encode: [150, 516, 850]
- Input str: '150,516,850,'
- Input str token list: ['150', ',', '516', ',', '850', ',']
- Input str token Id list: [3965, 11, 20571, 11, 16217, 11]

Sampling: *Softmax* over the digits tokens

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

Outline

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 θ:

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

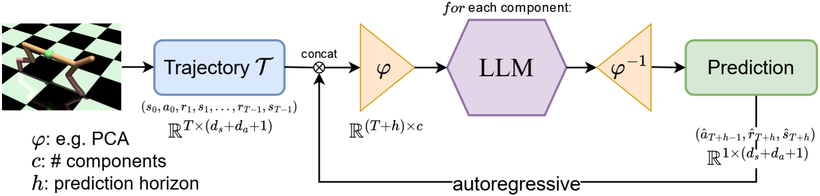
Challenges:

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

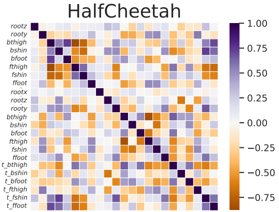
Outline

DICL

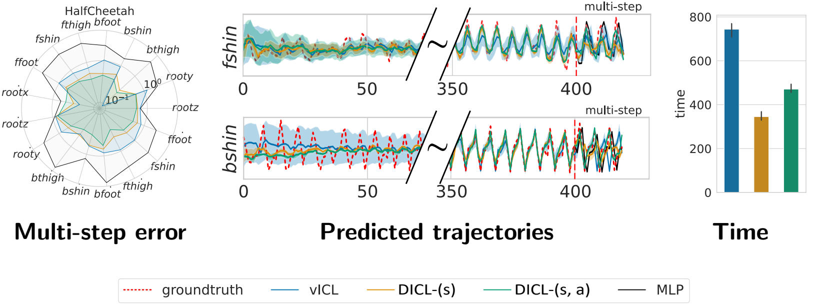
DICL: Disentangled In-Context Learning [1]



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



Outline



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)

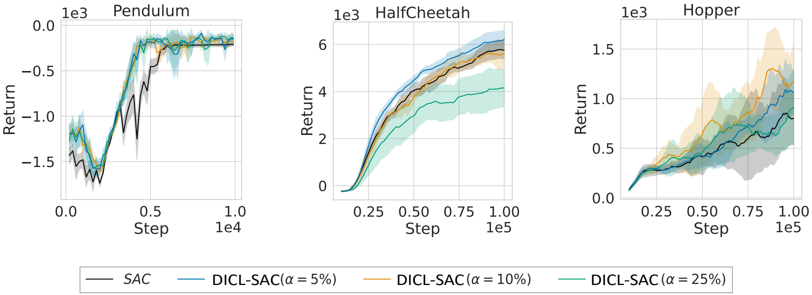
$+$

DICL

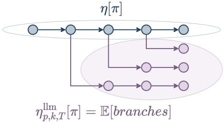
$=$

DICL-SAC

for $t = 1, \ldots, N\_interactions$ **do**
    New transition (s_t, a_t, r_t, s_{t+1}) from π_θ
    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 $\mathcal{R}$
        $\{\hat{s}_{i+1}\}_{0 \le i \le T_{\max}} \sim$ DICL-(s) $(\mathcal{T})$
        Add $\{(s_i, \tilde{a}_i, r_i, \hat{s}_{i+1})\}_{T \le i \le T_{\max}}$ 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 ϕ and ψ on $\mathcal{B} \cup \mathcal{B}_{\text{llm}}$
    end if
end for



Under mild assumptions on the LLM prediction error ε_{llm}, we have:



Theorem (Multi-branch return bound)

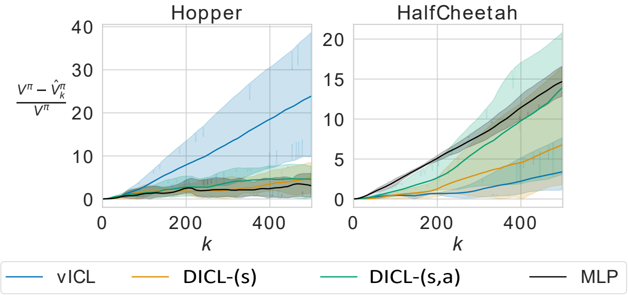
- *T the context length*
- *$p \in [0, 1]$ probability of branching*
- *k the branch length*
- *ε_{llm} the LLM in-context learning prediction error*

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

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

Results: Policy evaluation

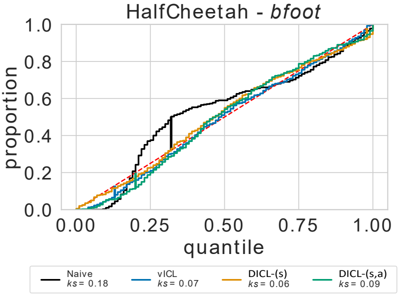
We levarage **DICL** for **Policy evaluation** by predicting the rewards trajectory in-context.



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.



Conclusion

- We presented **DICL**, a methodology to adapt LLMs for the task of dynamics learning in MBRL.
- We leveraged **DICL** for data-augmented RL, policy evaluation, and showed that LLMs are well-calibrated.

Take Home Message

LLMs are powerful foundation models trained on vast amounts of data
$\rightarrow$ **DICL** is an effective way to adapt them to MBRL

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.

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/dicl.pdf