Tabular Foundation Models

onepoint. x Telecom AI - 2026/06/02

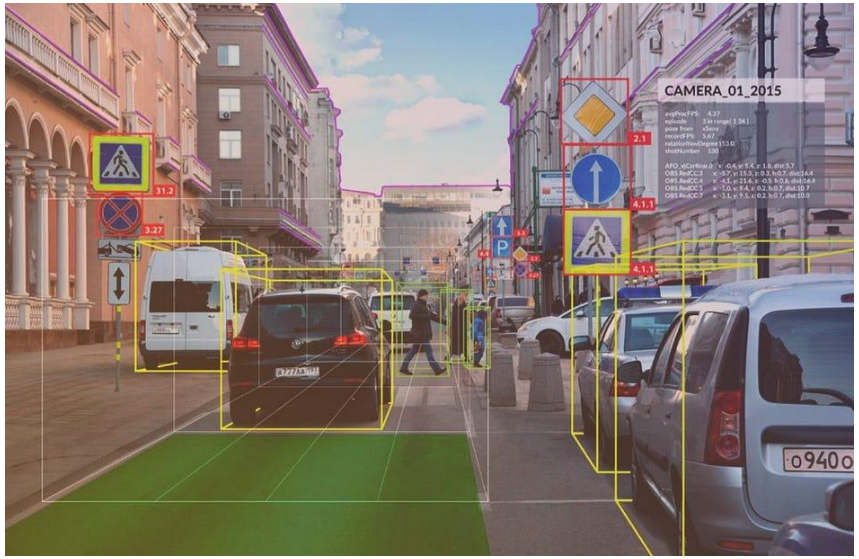
Données tabulaires et deep learning

Le Machine Learning et le Deep Learning ont révolutionné les domaines de la vision, du NLP, etc.

Mais pas le domaine des données tabulaires. Les modèles de type Random Forest ou XGBoost restent privilégiés.

Quelle différence y a-t-il entre les données tabulaires et les autres modalités ?

Pourquoi le *deep learning* ne fonctionne-t-il pas sur les données tabulaires ?



Patient ID	L-Carnitine	Creatinine	Pre-existing conditions	Homocysteine	Beta-hydroxybutyrate	Early stage Alzheimers?
P0001	1.2	15	None	3.5	1.8	No
P0002	1.3	13	None	3.4	1.8	No
P0003	1.2	14	Diabetes	n/a	1.7	Yes
...	...	...	...	...	...	...
P5001	3.2	30	None	4	3	?

Tabular Predicton Task

La complexité des données tabulaires

Le ratio #samples / #features :

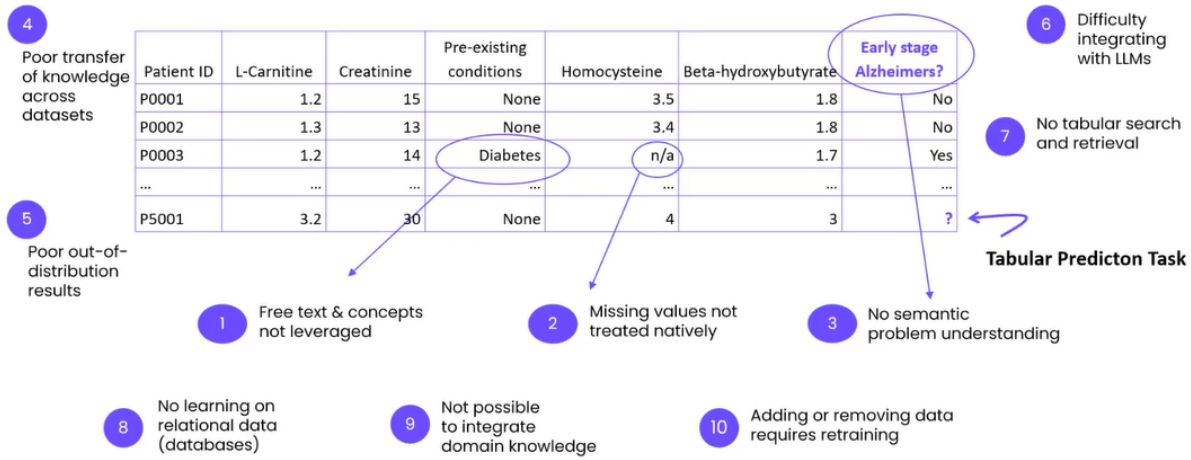
Les données tabulaires contiennent généralement beaucoup d'échantillons pour assez peu de *features*.
Comparaison avec la vision : chaque image de 100*100 pixels avec 3 channels (RGB) est représentée par 30,000 features.

Des données hétérogènes :

Des données continues mixées avec des données catégorielles.
Diverses modalités.
Des échelles variées.

Le problème de l'interprétabilité :

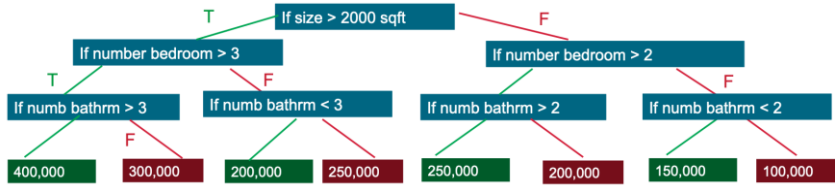
Les colonnes ont une sémantique et on souhaite souvent pouvoir évaluer leur impact sur une prédiction.
Les modèles « boîtes noires » sont inadaptés.
Comparaison avec la vision : le rôle individuel d'un pixel importe peu.



Random forests / XGBoost

Arbres de décision

Prédit l'étiquette (label) en évaluant un arbre de questions vrai/faux de type « *if-then-else* » portant sur les variables (features).

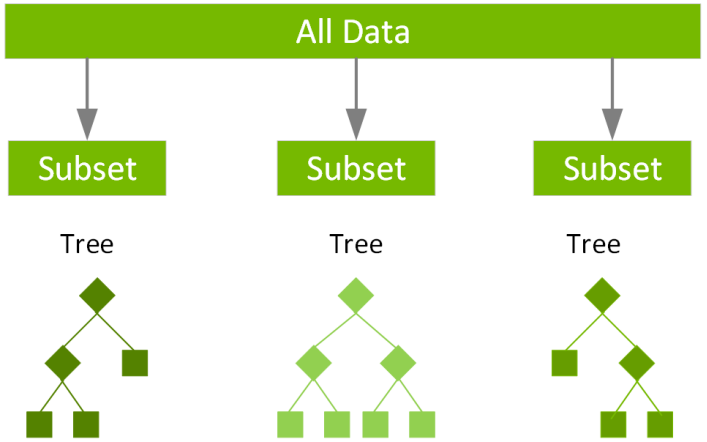


Bagging (*Random Forests*)

Construit des arbres de décision complets en parallèle à partir de sous-ensembles d'échantillons aléatoires issus du jeu de données. La prédiction finale est une moyenne de toutes les prédictions des arbres de décision.

Boosting

Combine plusieurs modèles faibles individuellement (*weak learners*) pour en faire un apprenant fort (*strong learner*). Chaque modèle faible est entraîné afin de corriger les erreurs commises par les modèles précédents.



Gradient boosting

Extension du boosting dans laquelle le processus de génération additive de modèles faibles est formalisé comme un algorithme de descente de gradient sur une fonction objectif. Le gradient boosting définit des cibles (basées sur le gradient de l'erreur par rapport à la prédiction) pour le modèle suivant, dans le but de minimiser les erreurs.

Decision trees + Gradient Boosting (ex: XGBoost)

Entraînent de manière itérative un ensemble d'arbres de décision peu profonds, chaque itération utilisant les erreurs résiduelles du modèle précédent pour ajuster le modèle suivant. La prédiction finale est une somme pondérée de toutes les prédictions des arbres.

Tabular Foundation Models

- **Bayesian Inference**
- PFN (Prior-data fitted networks)
- In-context learning
- Priors on datasets

TabICLv2: A better, faster, scalable, and open tabular foundation model

Jingang Qu [*1] David Holzmüller [*1] Gaël Varoquaux [1 2] Marine Le Morvan [1]

Abstract

Tabular foundation models, such as TabPFNv2 and TabICL, have recently dethroned gradient-boosted trees at the top of predictive benchmarks, demonstrating the value of in-context learning for tabular data. We introduce TabICLv2, a new state-of-the-art foundation model for regression and classification built on three pillars: (1) a novel synthetic data generation engine designed for high pretraining diversity; (2) various architectural innovations, including a new scalable softmax in attention improving generalization to larger datasets without prohibitive long-sequence pretraining; and (3) optimized pretraining protocols, notably replacing AdamW with the Muon optimizer. On the TabArena and TALENT benchmarks, TabICLv2 without any tuning surpasses the performance of the current state of the art, RealTabPFN-2.5 (hyperparameter-tuned, ensembled, and fine-tuned on real data). With only moderate pretraining compute, TabICLv2 generalizes effectively to million-scale datasets under 50GB GPU memory while being markedly faster than

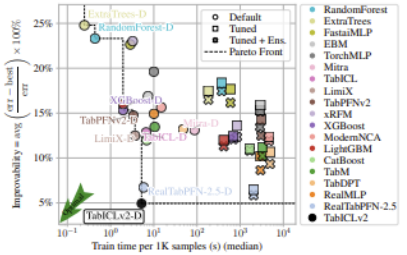


Figure 1. **Improvability vs. train time on TabArena (Erickson et al., 2025).** Improvability (lower is better) measures the relative error gap to the best method, averaged across datasets. Train time is training + inference in 8-fold cross-validation. For foundation models, it is dominated by forward passes that perform in-context learning. *Default* uses default hyperparameters; *Tuned* selects the best of 200 random hyperparameter configurations on validation; *Tuned + Ens.* applies post-hoc weighted ensemble of all configurations. The runtime of TabICLv2 is measured on an H100 GPU, while others are from TabArena. Results for inapplicable model-dataset pairs are imputed with default RandomForest.

PRIOR

TabPFN-3: Technica

Prior Labs Team (see Appendix A

Tabular data underpins most high-value
has driven the foundation model revol
users, TabPFN-3 builds on this founda
1M training rows and substantially red
synthetic data from our prior, TabPF
and brings substantial gains on time s

Model Development & Deployment. Noah Hollmann, Frank Hutter, Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin (Liam) Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing

Distribution & Product. Sauraj Gambhir, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova

Operations. Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen

Authors are ordered by their date of joining Prior Labs; all authors above affiliated with Prior Labs at the time of contribution; work done at Prior Labs.

Scientific Advisors. Samuel Müller, Madelon Hulsebos, Yann LeCun, Bernhard Schölkopf

Scientific advisors did not contribute IP.

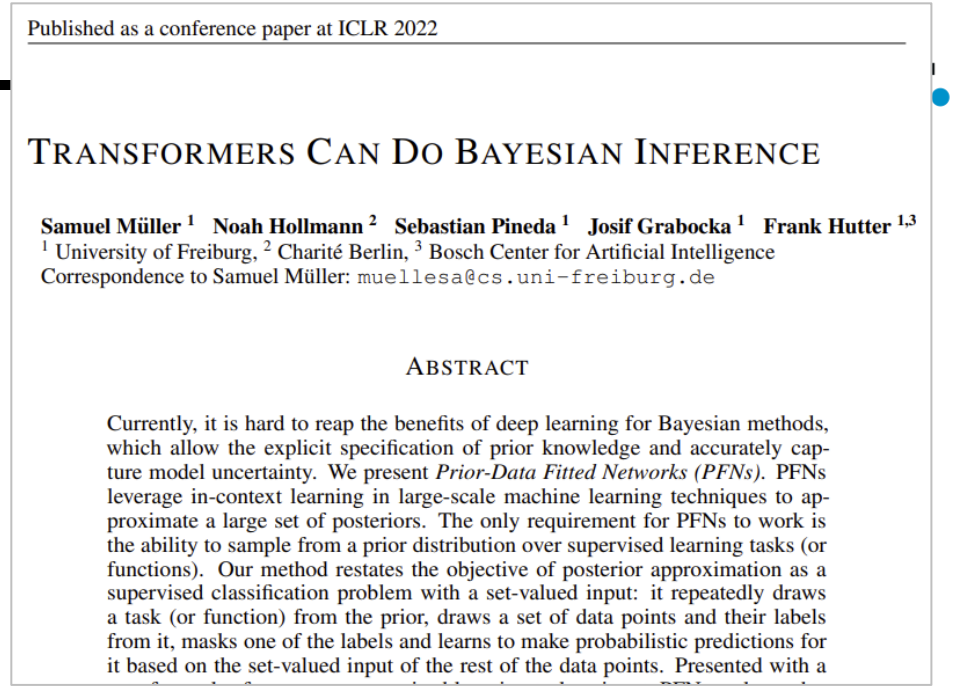
Bayesian Inference

ML supervisé ou non supervisé

- À partir d'une liste d'observations on souhaite faire des prédictions pour des observations jamais vues

$$D_{\text{train}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \qquad p(\mathbf{x}),\ \mathbf{x} \in D_{\text{test}} \not\subset D_{\text{train}}$$

$$D_{\text{train}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \qquad p(y|\mathbf{x}),\ (\mathbf{x}, y) \in D_{\text{test}} \not\subset D_{\text{train}}$$



Published as a conference paper at ICLR 2022

TRANSFORMERS CAN DO BAYESIAN INFERENCE

Samuel Müller [1] **Noah Hollmann** [2] **Sebastian Pineda** [1] **Josif Grabocka** [1] **Frank Hutter** [1,3]
[1] University of Freiburg, [2] Charité Berlin, [3] Bosch Center for Artificial Intelligence
Correspondence to Samuel Müller: muellesa@cs.uni-freiburg.de

ABSTRACT

Currently, it is hard to reap the benefits of deep learning for Bayesian methods, which allow the explicit specification of prior knowledge and accurately capture model uncertainty. We present *Prior-Data Fitted Networks (PFNs)*. PFNs leverage in-context learning in large-scale machine learning techniques to approximate a large set of posteriors. The only requirement for PFNs to work is the ability to sample from a prior distribution over supervised learning tasks (or functions). Our method restates the objective of posterior approximation as a supervised classification problem with a set-valued input: it repeatedly draws a task (or function) from the prior, draws a set of data points and their labels from it, masks one of the labels and learns to make probabilistic predictions for it based on the set-valued input of the rest of the data points. Presented with a

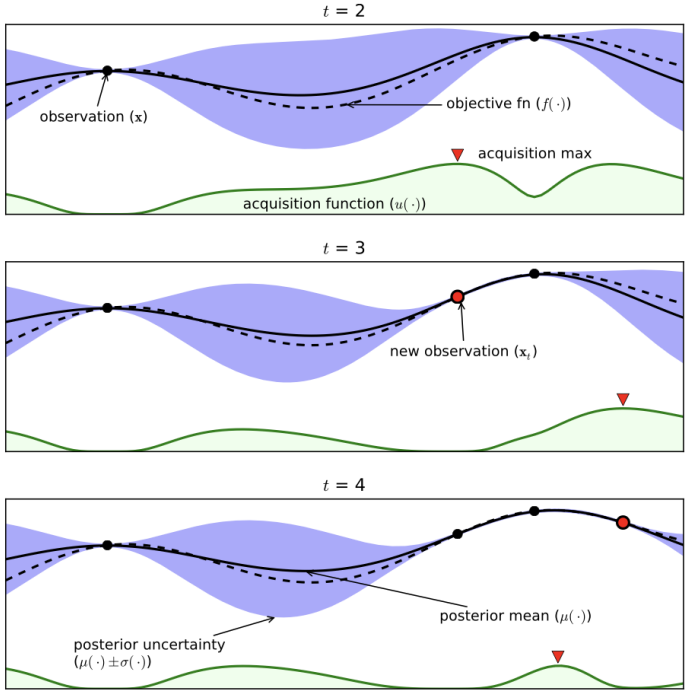
- Pour éviter le problème du ***No Free Lunch Theorem***, il faut présupposer qu'elles ont une certaine structure définie par une distribution a priori ≡ **prior** ≡ biais inductif.
- Une possibilité est de modéliser cette *prior* par une distribution de probabilité $p(\boldsymbol{\phi})$ sur les paramètres (variables latentes) $\boldsymbol{\phi}$ d'un modèle génératif $p(D|\boldsymbol{\phi})$ des données. L'observation d'un train set D_{train} permet alors de calculer la distribution a posteriori sur ces variables latentes

$$\underbrace{p(\boldsymbol{\phi}|D_{\text{train}})}_{\text{posterior}} = \frac{1}{Z} \underbrace{p(D_{\text{train}}|\boldsymbol{\phi})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\phi})}_{\text{prior}}$$

Bayesian Inference

La **distribution prédictive** PPD (pour le ML supervisé p.ex.) s'obtient alors par

$$p(y|\mathbf{x}, D_{\text{train}}) = \int p(y|\mathbf{x}, \boldsymbol{\phi})\, p(\boldsymbol{\phi}|D_{\text{train}})\, \mathrm{d}\boldsymbol{\phi}$$
$$\propto \int p(y|\mathbf{x}, \boldsymbol{\phi})\, p(D_{\text{train}}|\boldsymbol{\phi})\, p(\boldsymbol{\phi})\, \mathrm{d}\boldsymbol{\phi}$$



Connexions avec le ML supervisé usuel :

Approximation du maximum de vraisemblance :

$$p(\boldsymbol{\phi}|D_{\text{train}}) \to \delta(\boldsymbol{\phi} - \boldsymbol{\phi}_{\text{ML}}), \quad \boldsymbol{\phi}_{\text{ML}} = \arg\max_{\boldsymbol{\phi}} p(D_{\text{train}}|\boldsymbol{\phi})$$

Approximation de maximum a posteriori :

$$p(\boldsymbol{\phi}|D_{\text{train}}) \to \delta(\boldsymbol{\phi} - \boldsymbol{\phi}_{\text{MAP}}), \quad \boldsymbol{\phi}_{\text{MAP}} = \arg\max_{\boldsymbol{\phi}} p(D_{\text{train}}|\boldsymbol{\phi})\, p(\boldsymbol{\phi})$$

Tabular Foundation Models

- Bayesian Inference
- **PFN (Prior-data fitted networks)**
- In-context learning
- Priors on datasets

TabICLv2: A better, faster, scalable, and open tabular foundation model

Jingang Qu [*1] David Holzmüller [*1] Gaël Varoquaux [1 2] Marine Le Morvan [1]

Abstract

Tabular foundation models, such as TabPFNv2 and TabICL, have recently dethroned gradient-boosted trees at the top of predictive benchmarks, demonstrating the value of in-context learning for tabular data. We introduce TabICLv2, a new state-of-the-art foundation model for regression and classification built on three pillars: (1) a novel synthetic data generation engine designed for high pretraining diversity; (2) various architectural innovations, including a new scalable softmax in attention improving generalization to larger datasets without prohibitive long-sequence pretraining; and (3) optimized pretraining protocols, notably replacing AdamW with the Muon optimizer. On the TabArena and TALENT benchmarks, TabICLv2 without any tuning surpasses the performance of the current state of the art, RealTabPFN-2.5 (hyperparameter-tuned, ensembled, and fine-tuned on real data). With only moderate pretraining compute, TabICLv2 generalizes effectively to million-scale datasets under 50GB GPU memory while being markedly faster than

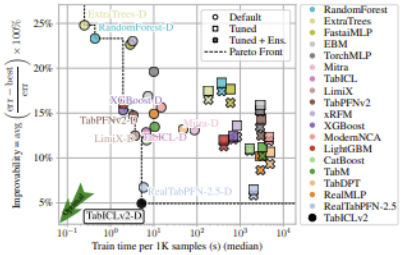


Figure 1. **Improvability vs. train time on TabArena (Erickson et al., 2025).** Improvability (lower is better) measures the relative error gap to the best method, averaged across datasets. Train time is training + inference in 8-fold cross-validation. For foundation models, it is dominated by forward passes that perform in-context learning. *Default* uses default hyperparameters; *Tuned* selects the best of 200 random hyperparameter configurations on validation; *Tuned + Ens.* applies post-hoc weighted ensemble of all configurations. The runtime of TabICLv2 is measured on an H100 GPU, while others are from TabArena. Results for inapplicable model-dataset pairs are imputed with default RandomForest.

PRIOR

TabPFN-3: Technica

Prior Labs Team (see Appendix A

Tabular data underpins most high-value
has driven the foundation model revol
users, TabPFN-3 builds on this founda
1M training rows and substantially red
synthetic data from our prior, TabPFN
and brings substantial gains on time s

Model Development & Deployment. Noah Hollmann, Frank Hutter, Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin (Liam) Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing

Distribution & Product. Sauraj Gambhir, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova

Operations. Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen

Authors are ordered by their date of joining Prior Labs; all authors above affiliated with Prior Labs at the time of contribution; work done at Prior Labs.

Scientific Advisors. Samuel Müller, Madelon Hulsebos, Yann LeCun, Bernhard Schölkopf

Scientific advisors did not contribute IP.

Prior-data Fitted Networks (PFN)

On va échantillonner des exemples de datasets $D \sim p(D)$ et des couples $(\mathbf{x}, y)$ puis essayer d'apprendre directement la distribution prédictive $(\mathbf{x}, D) \longrightarrow p(y|\mathbf{x}, D)$ à l'aide d'un modèle paramétrique (un PFN) $(\mathbf{x}, D) \longrightarrow q_\theta(y|\mathbf{x}, D)$ et d'une fonction de coût $\ell(\theta)$ appropriée.

$$
\begin{aligned}
\ell(\theta) &:= \mathbb{E}_{(\mathbf{x},y)\cup D_n \sim p(D_{n+1})}\left[-\log q_\theta(y|\mathbf{x}, D_n)\right] \\
&= -\sum_{\mathbf{x},y,D_n} p(\mathbf{x}, y, D_n)\ \log q_\theta(y|\mathbf{x}, D_n) \\
&= -\sum_{\mathbf{x},D_n} p(\mathbf{x}, D_n) \sum_y p(y|\mathbf{x}, D_n)\ \log q_\theta(y|\mathbf{x}, D_n) \\
&= -\sum_{\mathbf{x},D_n} p(\mathbf{x}, D_n) \sum_y p(y|\mathbf{x}, D_n) \left(\log \frac{q_\theta(y|\mathbf{x}, D_n)}{p(y|\mathbf{x}, D_n)} + \log p(y|\mathbf{x}, D_n)\right) \\
&= \sum_{\mathbf{x},D_n} p(\mathbf{x}, D_n) \left(\mathrm{KL}\left[p(\cdot|\mathbf{x}, D_n) \| q_\theta(\cdot|\mathbf{x}, D_n)\right] + H\left[p(\cdot|\mathbf{x}, D_n)\right]\right) \\
&= \mathbb{E}_{\mathbf{x},D_n\sim p}\ \mathrm{KL}\left[p(\cdot|\mathbf{x}, D_n) \| q_\theta(\cdot|\mathbf{x}, D_n)\right] \ + \text{const}
\end{aligned}
$$

Minimiser $\ell(\theta)$ revient donc à chercher une approximation $q_{\theta^*}(y|\mathbf{x}, D)$ de la vraie distribution prédictive $p(y|\mathbf{x}, D)$.

S'il existe un θ^* pour lequel $q_{\theta^*} = p$ on le trouvera en minimisant $\ell(\theta)$.

Pas d'overfitting car plus KL est proche de zéro, plus q_θ est proche du vrai p !

Prior-data Fitted Networks (PFN)

Atout de PFN : il suffit d'avoir un mécanisme efficace d'échantillonnage des jeux de données $D_n \sim p(D_n)$

Algorithm 1: Training a PFN model by Fitting Prior-Data

Input : A prior distribution over datasets $p(\mathcal{D})$, from which samples can be drawn and the number of samples K to draw

Output : A model q_θ that will approximate the PPD

Initialize the neural network q_θ;

for $j \leftarrow 1$ **to** K **do**

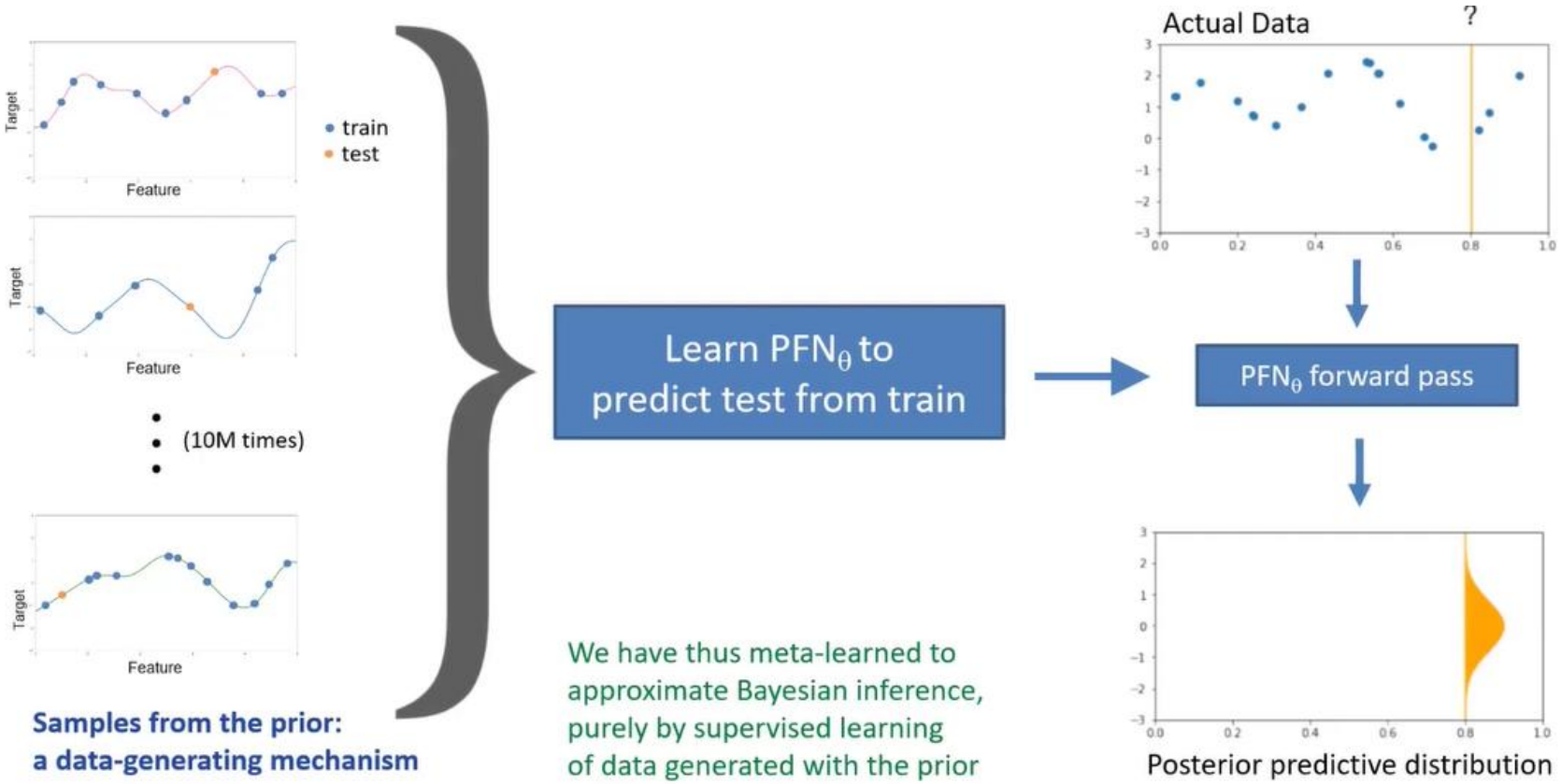
  Sample $D \cup \{(x_i, y_i)\}_{i=1}^{m} \sim p(\mathcal{D})$;

  Compute stochastic loss approximation $\bar{\ell}_\theta = \sum_{i=1}^{m}(-\log q_\theta(y_i|x_i, D))$;

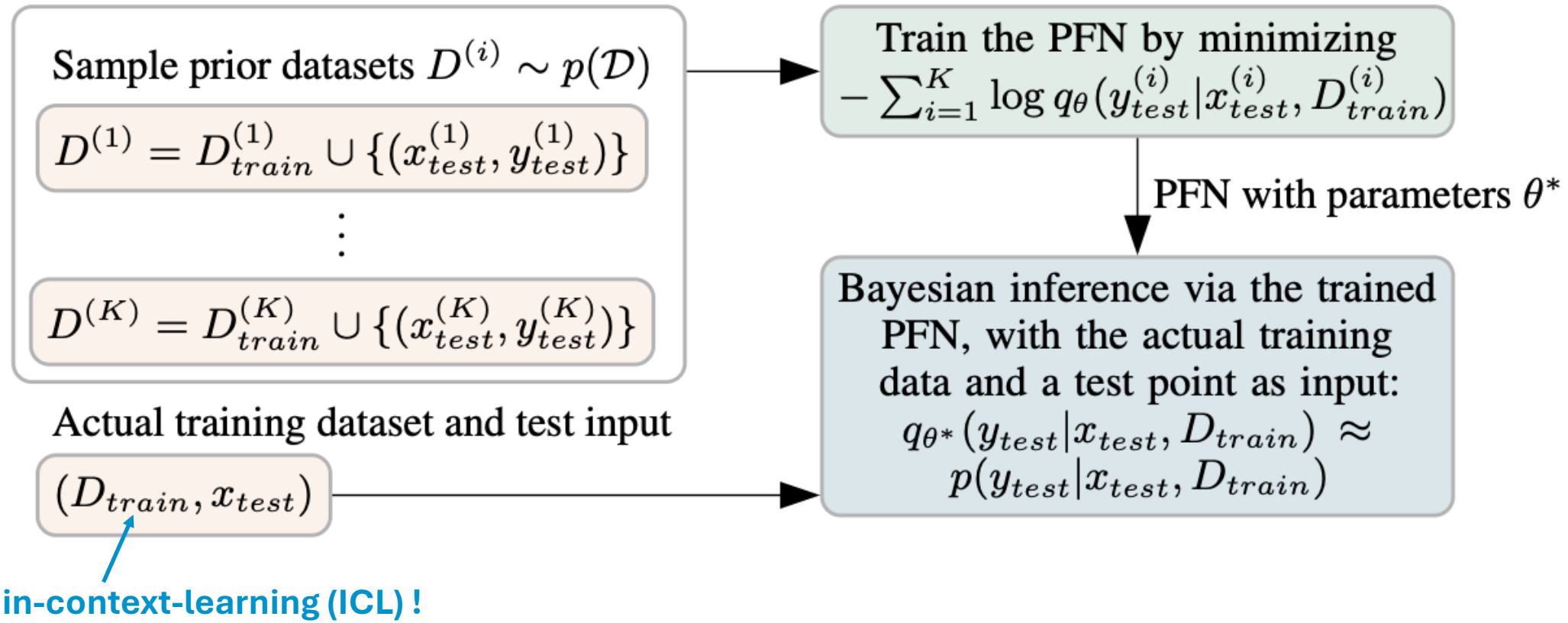
  Update parameters θ with stochastic gradient descent on $\nabla_\theta \bar{\ell}_\theta$;

end

Prior-data Fitted Networks (PFN)



Prior-data Fitted Networks (PFN)



Tabular Foundation Models

- Bayesian Inference
- PFN (Prior-data fitted networks)
- **In-context learning**
- Priors on datasets

TabICLv2: A better, faster, scalable, and open tabular foundation model

Jingang Qu [*1] David Holzmüller [*1] Gaël Varoquaux [1 2] Marine Le Morvan [1]

Abstract

Tabular foundation models, such as TabPFNv2 and TabICL, have recently dethroned gradient-boosted trees at the top of predictive benchmarks, demonstrating the value of in-context learning for tabular data. We introduce TabICLv2, a new state-of-the-art foundation model for regression and classification built on three pillars: (1) a novel synthetic data generation engine designed for high pretraining diversity; (2) various architectural innovations, including a new scalable softmax in attention improving generalization to larger datasets without prohibitive long-sequence pretraining; and (3) optimized pretraining protocols, notably replacing AdamW with the Muon optimizer. On the TabArena and TALENT benchmarks, TabICLv2 without any tuning surpasses the performance of the current state of the art, RealTabPFN-2.5 (hyperparameter-tuned, ensembled, and fine-tuned on real data). With only moderate pretraining compute, TabICLv2 generalizes effectively to million-scale datasets under 50GB GPU memory while being markedly faster than

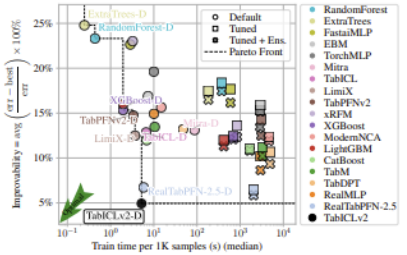


Figure 1. **Improvability vs. train time on TabArena (Erickson et al., 2025).** Improvability (lower is better) measures the relative error gap to the best method, averaged across datasets. Train time is training + inference in 8-fold cross-validation. For foundation models, it is dominated by forward passes that perform in-context learning. *Default* uses default hyperparameters; *Tuned* selects the best of 200 random hyperparameter configurations on validation; *Tuned + Ens.* applies post-hoc weighted ensemble of all configurations. The runtime of TabICLv2 is measured on an H100 GPU, while others are from TabArena. Results for inapplicable model-dataset pairs are imputed with default RandomForest.

PRIOR

TabPFN-3: Technica

Prior Labs Team (see Appendix A

Tabular data underpins most high-value
has driven the foundation model revol
users, TabPFN-3 builds on this founda
1M training rows and substantially red
synthetic data from our prior, TabPFN
and brings substantial gains on time s

Model Development & Deployment. Noah Hollmann, Frank Hutter, Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin (Liam) Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing

Distribution & Product. Sauraj Gambhir, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova

Operations. Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen

Authors are ordered by their date of joining Prior Labs; all authors above affiliated with Prior Labs at the time of contribution; work done at Prior Labs.

Scientific Advisors. Samuel Müller, Madelon Hulsebos, Yann LeCun, Bernhard Schölkopf

Scientific advisors did not contribute IP.

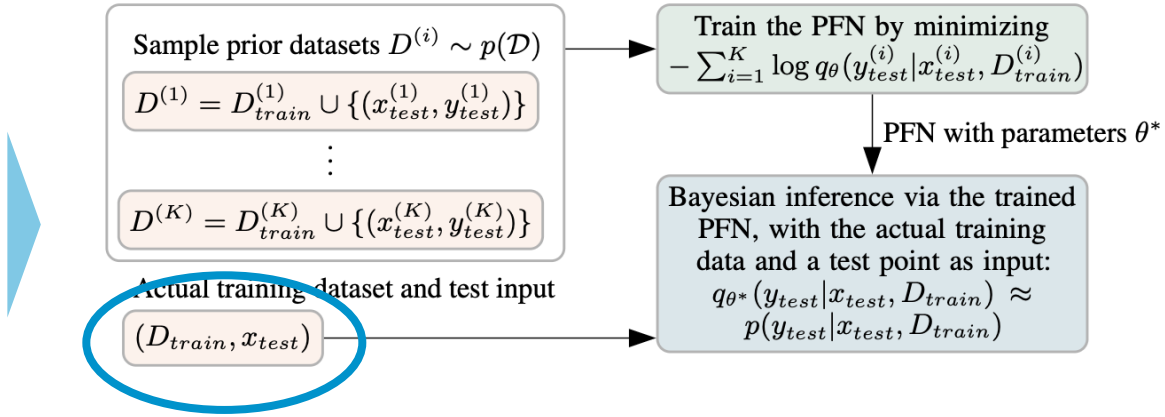
In-context learning

L'apprentissage en contexte consiste à entraîner un modèle de langage de grande envergure (LLM) à partir d'une instruction comprenant un ensemble d'exemples (paires entrée/sortie ou exemples en contexte) généralement rédigés en langage naturel et intégrés à la séquence d'entrée. Ces exemples, souvent tirés d'un ensemble de données, ne servent pas à réentraîner le modèle, mais sont directement intégrés à sa fenêtre de contexte.

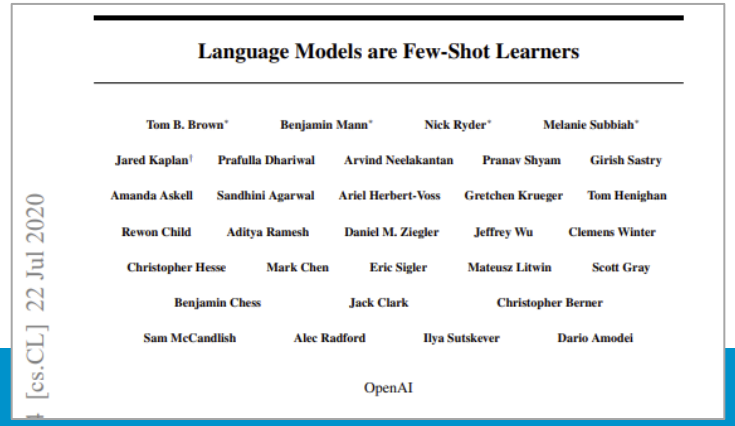
LLM input:

Review: The movie was fantastic → Sentiment: Positive
Review: I hated the storyline → Sentiment: Negative
Review: The music was pleasant → Sentiment:

LLM output:
Positive



Zero-shot prompting: la tâche est expliquée sans fournir d'exemples.
One-shot prompting: un seul exemple est fourni pour illustrer la tâche.
Few-shot prompting: plusieurs exemples sont fournis.
Chain-of-thought prompting: chaque exemple comprend des étapes de raisonnement intermédiaires pour guider la logique du modèle.

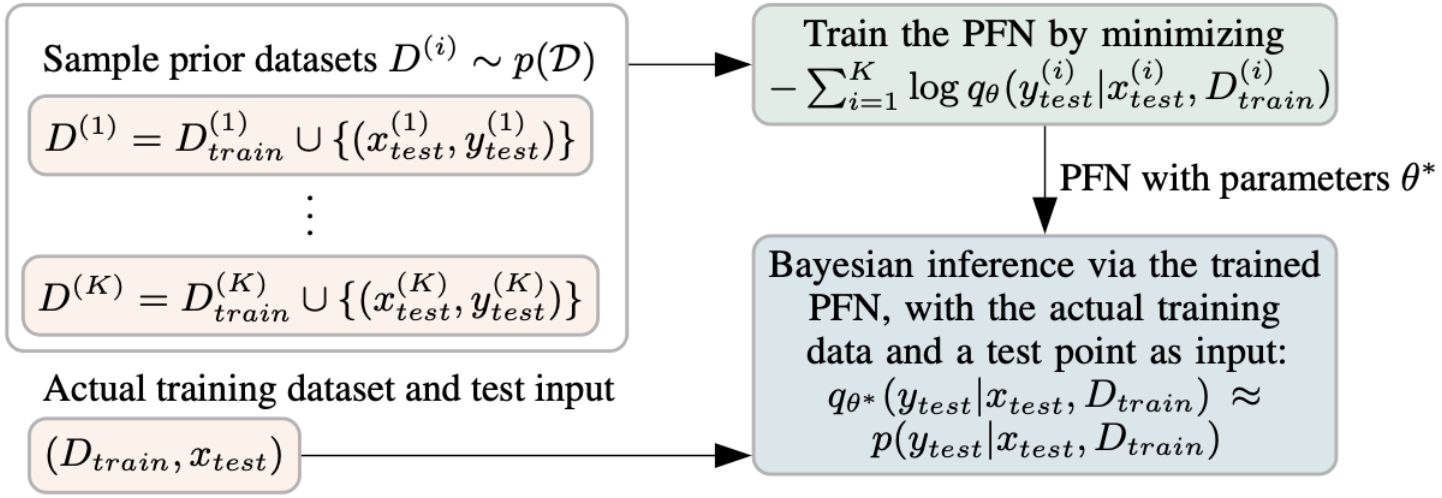

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

[cs.CL] 22 Jul 2020

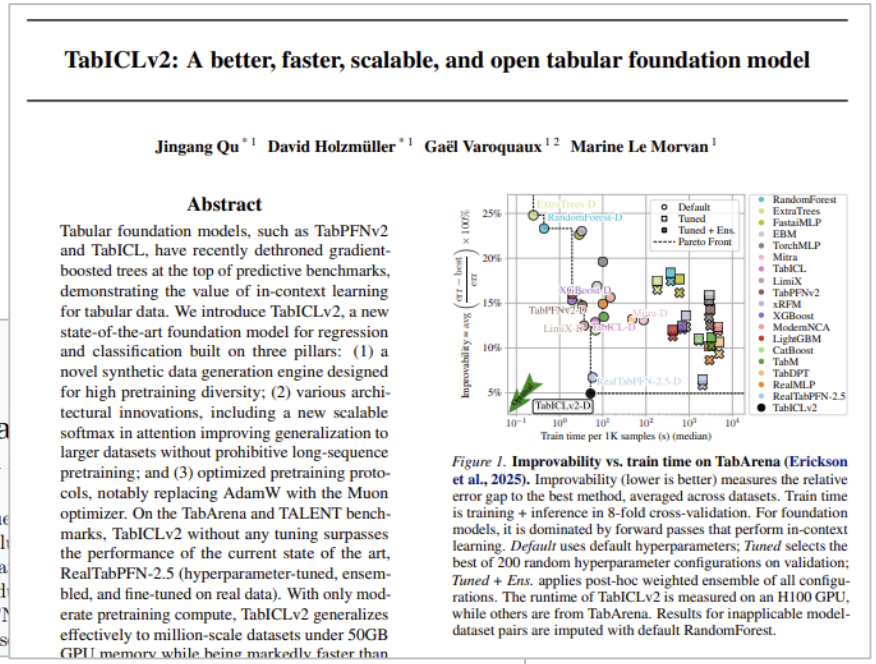
Implémentation et entraînement



- Nécessite beaucoup de calcul pour une unique prédiction
- Comment adapter le mécanisme d'attention aux données tabulaires ?
- Comment représenter les features ?
- **Comment échantillonner des datasets pour l'entraînement ?**

Tabular Foundation Models

- Bayesian Inference
- PFN (Prior-data fitted networks)
- In-context learning
- **Priors on datasets**



TabICLv2: A better, faster, scalable, and open tabular foundation model

Jingang Qu [*1] David Holzmüller [*1] Gaël Varoquaux [1 2] Marine Le Morvan [1]

Abstract

Tabular foundation models, such as TabPFNv2 and TabICL, have recently dethroned gradient-boosted trees at the top of predictive benchmarks, demonstrating the value of in-context learning for tabular data. We introduce TabICLv2, a new state-of-the-art foundation model for regression and classification built on three pillars: (1) a novel synthetic data generation engine designed for high pretraining diversity; (2) various architectural innovations, including a new scalable softmax in attention improving generalization to larger datasets without prohibitive long-sequence pretraining; and (3) optimized pretraining protocols, notably replacing AdamW with the Muon optimizer. On the TabArena and TALENT benchmarks, TabICLv2 without any tuning surpasses the performance of the current state of the art, RealTabPFN-2.5 (hyperparameter-tuned, ensembled, and fine-tuned on real data). With only moderate pretraining compute, TabICLv2 generalizes effectively to million-scale datasets under 50GB GPU memory while being markedly faster than

Figure 1. **Improvability vs. train time on TabArena (Erickson et al., 2025).** Improvability (lower is better) measures the relative error gap to the best method, averaged across datasets. Train time is training + inference in 8-fold cross-validation. For foundation models, it is dominated by forward passes that perform in-context learning. *Default* uses default hyperparameters; *Tuned* selects the best of 200 random hyperparameter configurations on validation; *Tuned + Ens.* applies post-hoc weighted ensemble of all configurations. The runtime of TabICLv2 is measured on an H100 GPU, while others are from TabArena. Results for inapplicable model-dataset pairs are imputed with default RandomForest.

PRIOR

TabPFN-3: Technica

Prior Labs Team (see Appendix A

Tabular data underpins most high-value
has driven the foundation model revol
users, TabPFN-3 builds on this founda
1M training rows and substantially red
synthetic data from our prior, TabPFN
and brings substantial gains on time s

Model Development & Deployment. Noah Hollmann, Frank Hutter, Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin (Liam) Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing

Distribution & Product. Sauraj Gambhir, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova

Operations. Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen

Authors are ordered by their date of joining Prior Labs; all authors above affiliated with Prior Labs at the time of contribution; work done at Prior Labs.

Scientific Advisors. Samuel Müller, Madelon Hulsebos, Yann LeCun, Bernhard Schölkopf

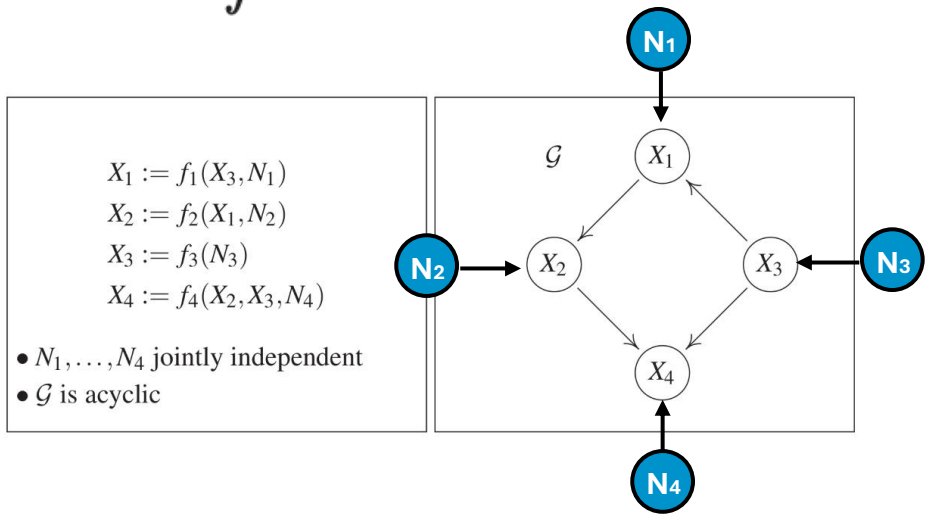
Scientific advisors did not contribute IP.

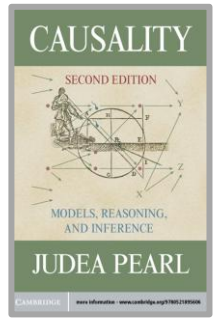
Échantillonner des datasets

Idée générale : générer des données qui ressemblent aux données du monde réel et qui proviennent donc de **mécanismes causaux** (physiques, sociaux, économiques,...)

$$q_\theta(y|\mathbf{x}, D) \approx p(y|\mathbf{x}, D) = \int p(y|\mathbf{x}, \boldsymbol{\phi}_{\text{SCM}})\, p(\boldsymbol{\phi}_{\text{SCM}}|D)\, \mathrm{d}\boldsymbol{\phi}_{\text{SCM}}$$

Recours aux DAG + SCM
(théorie de la causalité de Pearl).





Biometrika (1995), **82**, 4, *pp.* 669–710
Printed in Great Britain

Causal diagrams for empirical research (With Discussions)

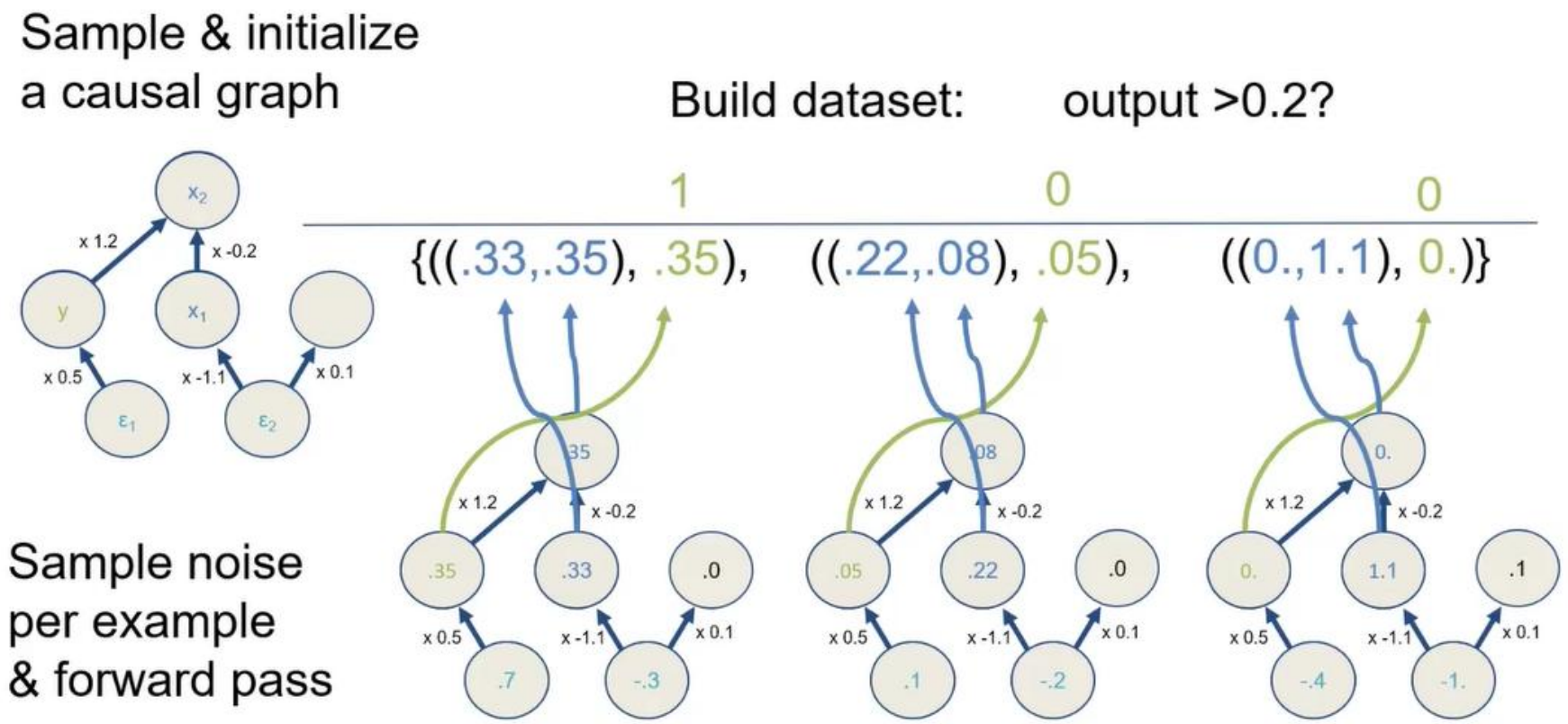
BY JUDEA PEARL

Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, California 90024, U.S.A.

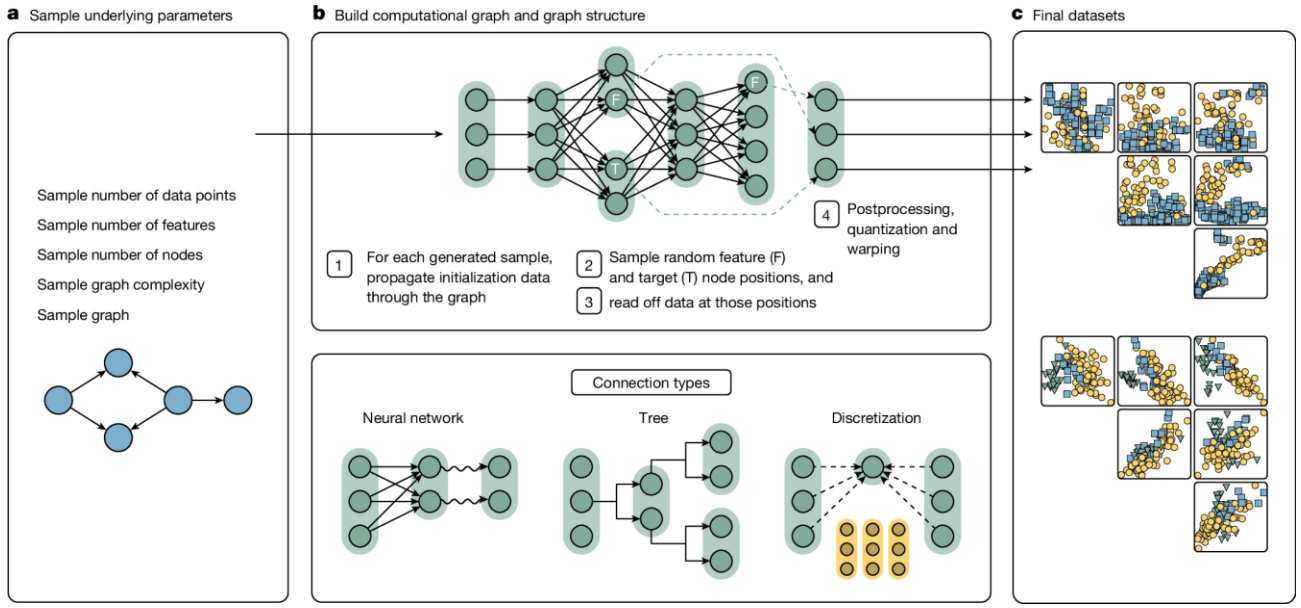
Les graphes (DAG) et les liens de ces SCM implémentent un **biais de simplicité** (Rasoir d'Occam) qui favorise les DAG avec peu de nœuds.

Les **hyperparamètres** des SCM sont définis par des **distributions.**

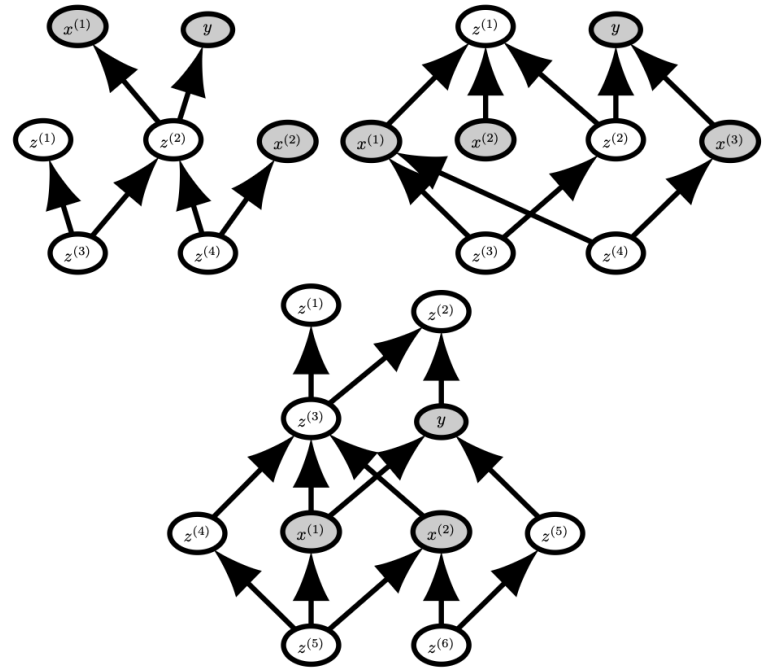
Échantillonner des datasets



Échantillonner des datasets

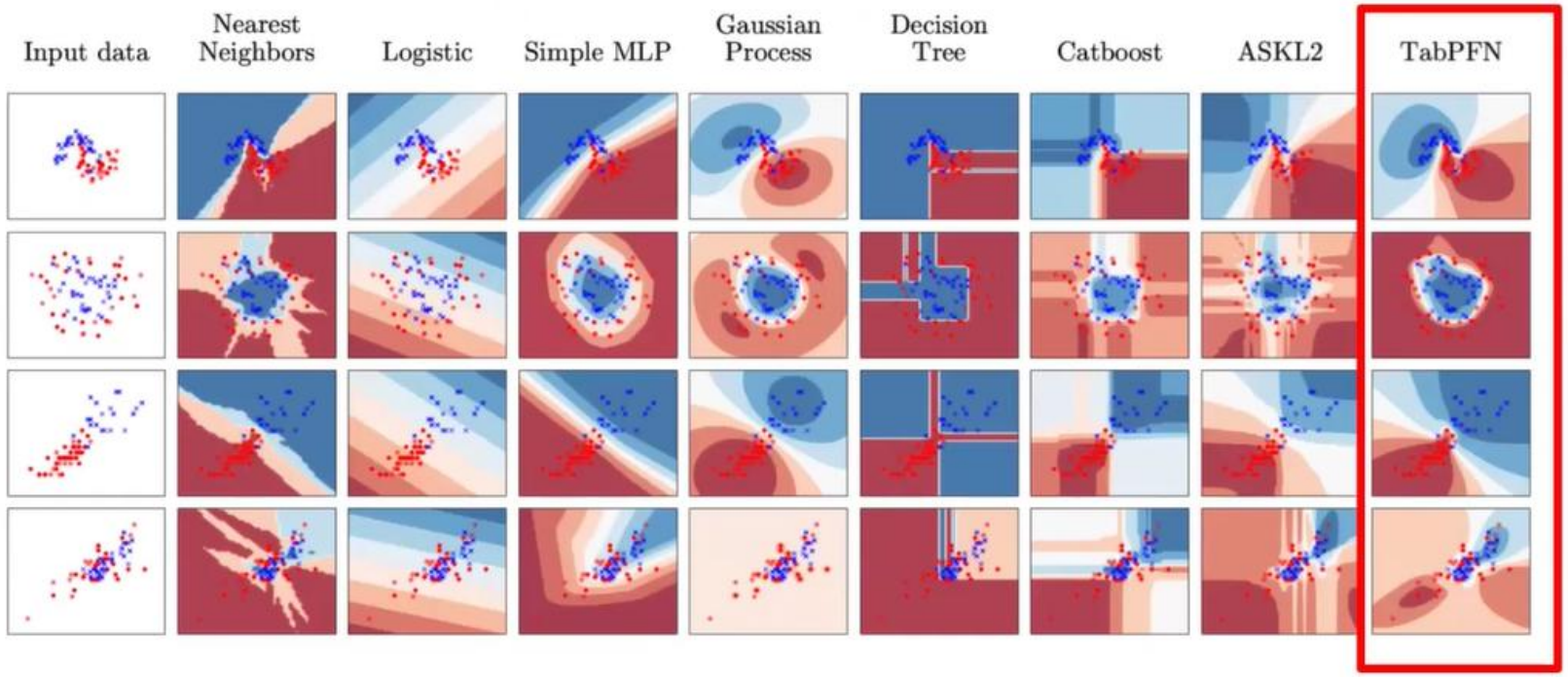


a, For each dataset, we first sample high-level hyperparameters. **b**, Based on these hyperparameters, we construct a structural causal model that encodes the computational function generating the dataset. Each node holds a vector and each edge in the computational graph implements a function according to one of the connection types. In step 1, using random noise variables we generate initialization data, which is fed into the root nodes of the graphs and propagated through the computational graph for each to-be-generated sample. In step 2, we randomly sample feature and target node positions in the graph, labelled F and T, respectively. In step 3, we extract the intermediate data representations at the sampled feature and target node positions. In step 4, we post-process the extracted data. **c,** We retrieve the final datasets. We plot interactions of feature pairs and the node colour represents the class of the sample.

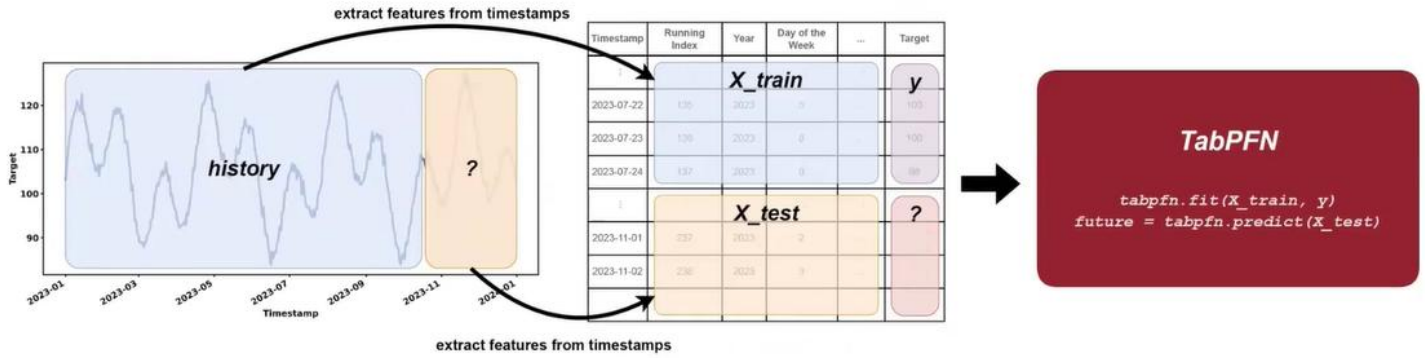


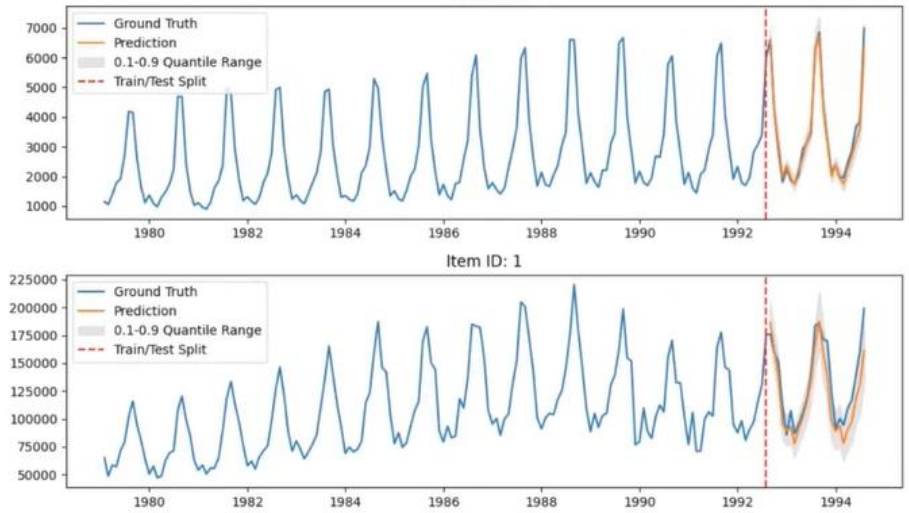
Données synthétiques vs. réelles

Prédictions calibrées



TabPFN et séries temporelles





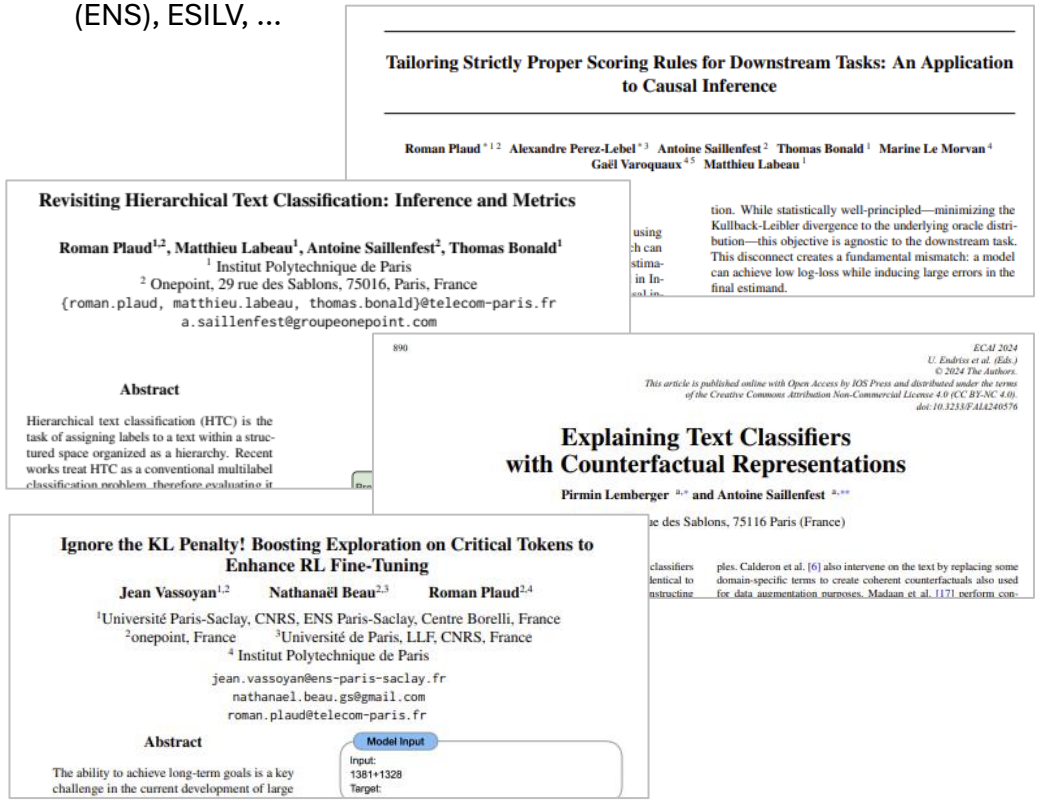
The End

Antoine SAILLENFEST
Chercheur chez onepoint
a.saillenfest@groupeonepoint.com
https://toinesayan.github.io/

Roman PLAUD
Doctorant CIFRE – onepoint x Télécom Paris
r.plaud@groupeonepoint.com
https://romanplaud.github.io/

TALia

Laboratoire de Traitement du langage interne à onepoint.
Initiative conjointe onepoint-Télécom Paris
Collaborations avec Université Paris Cité, Centre Borelli (ENS), ESILV, ...



Tailoring Strictly Proper Scoring Rules for Downstream Tasks: An Application to Causal Inference

Revisiting Hierarchical Text Classification: Inference and Metrics

Explaining Text Classifiers with Counterfactual Representations

Ignore the KL Penalty! Boosting Exploration on Critical Tokens to Enhance RL Fine-Tuning