Efficient Machine Learning for new physics discoveries

Gianvito Losapio

University of Genoa

Master Thesis in Computer Science

Supervisors
Prof. Lorenzo Rosasco
Marco Letizia, PhD

March 28th, 2022

Artificial intelligence (AI)

Machine Learning + Data

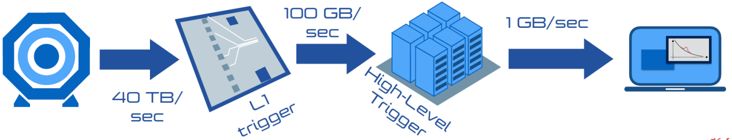
The costs of AI

GPT-3 (175B parameters)†:

- 355 years of GPU training
- $ 4.6 M

† https://lambdalabs.com/blog/demystifying-gpt-3/

Machine Learning for High Energy Physics



This thesis

Efficient Machine Learning for High Energy Physics

Model	Training time	Resources
Our approach	$\mathcal{O}(\mathrm{s})$	**Single GPU machine**
Neural networks	$\mathcal{O}(h)$	Farm of CPUs

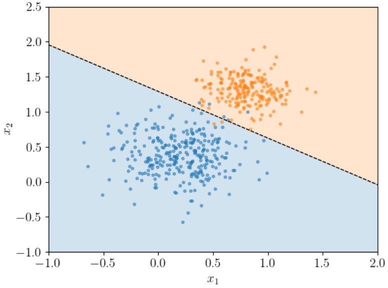
Supervised learning

Given a *training set*

$$(x_i, y_i)_{i=1}^{n} \in (\mathbb{R}^d \times \{0,1\})^n$$

find a *map*

$$f_n(x_{\text{new}}) \sim y_{\text{new}}$$



Loss & target functions

Empirical Risk Minimization: $$f_n = \arg\min \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$$

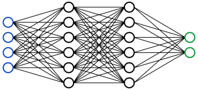
Logistic loss: $$\ell(y, f(x)) = \log(1 + e^{yf(x)}) \longrightarrow \lim_{n\to\infty} f_n = \log \frac{p(1|x)}{p(0|x)}$$

Models

$$f_w(x) = w^T x$$

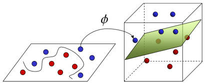
Neural networks

$$w^T \sigma(B^T x)$$

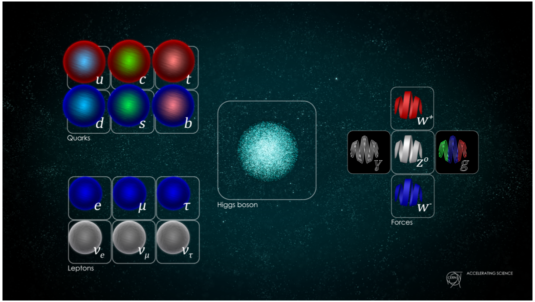


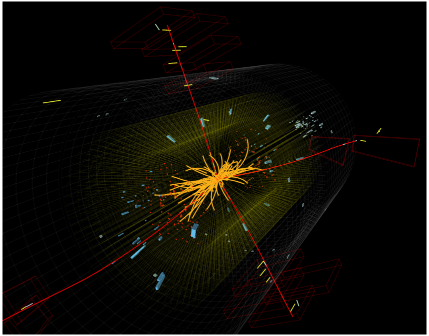
Kernel methods

$$w^T \Phi(x)$$



Theory vs Data





Theory vs Data

$$\{x_i\}_{i=1}^{\mathcal{N}_0} \sim p(x|0), N(0) \longleftarrow \text{ Standard Model (\textit{known})}$$

$$\{x_j\}_{j=1}^{\mathcal{N}_1} \sim p(x|1), N(1) \longleftarrow \text{ Data (\textit{true, unknown})}$$

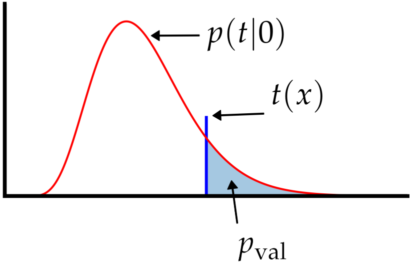
$$p(x \mid 0) = p(x \mid 1)\ ?$$

Idea hypothesis testing

1. $f_n(x) \approx \log \frac{p(x|1)}{p(x|0)} \longrightarrow t(x)$

2. toy experiments: $p(t|0)$

3. p-value: p_{val}



Remarks

- inspired by D'Agnolo et. al.[†]
- unbalanced problem with $\mathcal{N}_0 \gg \mathcal{N}_1$
- multiple training with $N_{\text{toy}} = \mathcal{O}(100)$

[†] D'Agnolo, R. T., & Wulzer, A. (2019). *Learning new physics from a machine.* Physical Review D, 99(1), 015014.

Weighted logistic loss

$$\ell(y, f(x)) = y\, \beta_1 \, \log\left(1 + e^{-f(x)}\right) + \, (1-y)\, \beta_0 \, \log\left(1 + e^{f(x)}\right)$$

$$\lim_{n \to \infty} f_n = \frac{\beta_1}{\beta_0} \frac{p(1|x)}{p(0|x)}$$

Physics-informed weights

$$\begin{cases} \beta_1 = \dfrac{\mathcal{N}_0}{N(0)} \\ \beta_0 = 1 \end{cases}$$

$$f_n \approx \log\left[\frac{n(x|1)}{n(x|0)}\right], \text{ with } n(x) = Np(x) \longrightarrow t(x)$$

Kernel logistic regression

$$\min_{w} L(f_w) + \lambda R(f_w)$$

$$f_w(x) = \sum_{i=1}^{n} w_i \, k(x, x_i), \quad \text{with} \quad k(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$$

LogFalkon†

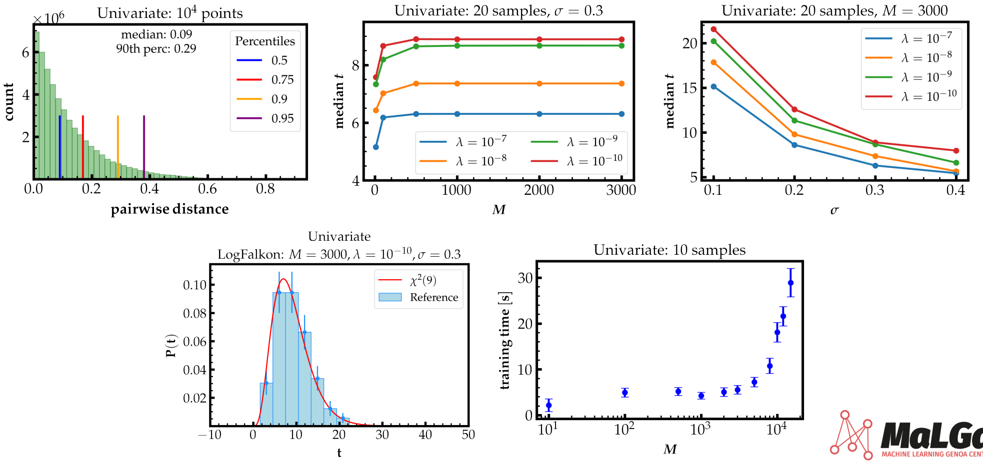
$$f_w(x) = \sum_{i=1}^{\mathbf{M}} w_i k(x, x_i)$$

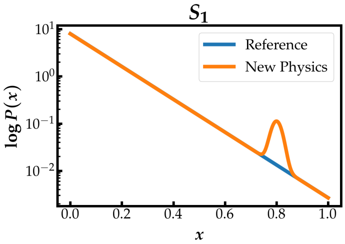
with $\{x_1, x_2, \ldots, x_{\mathbf{M}}\} \subseteq \{x_1, x_2, \ldots, x_n\}$ called Nystrom centers.

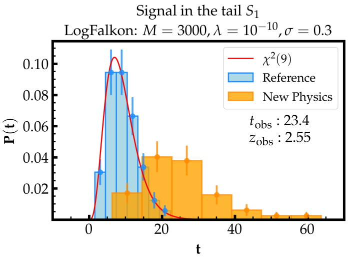
Tune: $\boxed{M, \lambda, \sigma}$

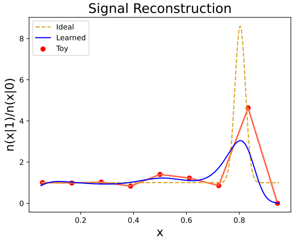
† Meanti, G., et. al. (2020). *Kernel methods through the roof: handling billions of points efficiently.* Advances in Neural Information Processing Systems, 33, 14410-14422.

Hyperparameter tuning

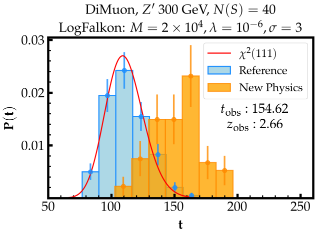


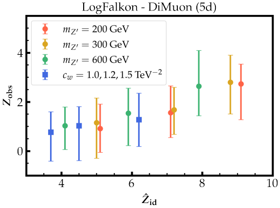


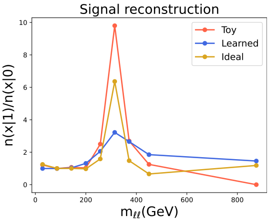


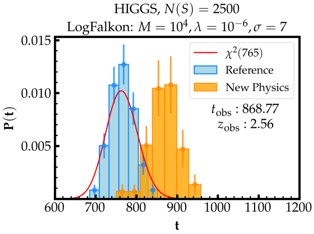


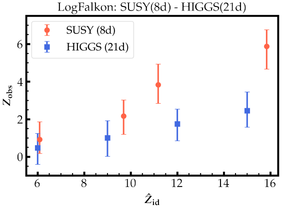
D'Agnolo, R. T., & Wulzer, A. (2019). *Learning new physics from a machine.* Physical Review D, 99(1), 015014.

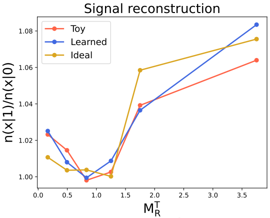






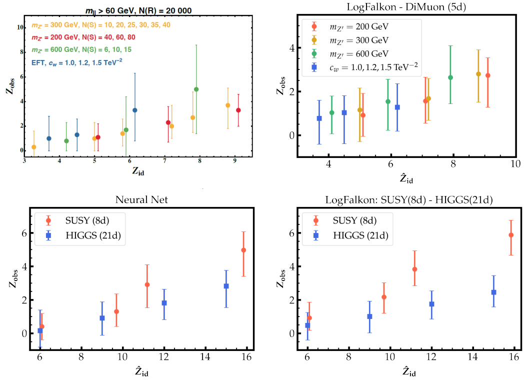






D'Agnolo, et. al. (2021). *Learning multivariate new physics.* The European Physical Journal C, 81(1), 1-21.
Baldi, P., et. al. (2014). *Searching for exotic particles in high-energy physics with deep learning.* Nature communications, 5(1), 1-9.

Comparison with neural networks



Model	DiMuon	SUSY	HIGGS
LogFalkon	**88.7 $\pm$ 1.9 s**	**44.8 $\pm$ 1.5 s**	**89.7 $\pm$ 2.2 s**
Neural networks	4.23 $\pm$ 0.73 h	73.1 $\pm$ 10 h	112 $\pm$ 9 h

MaLGa MACHINE LEARNING GENOA CENTER

Contributions

- An efficient ML framework for High Energy Physics
- A software library (Python)
- Extensive experiments

Future work

- Other datasets and domains
- Anomaly detection / Semi-supervised learning

Thank you!

Test statistic

$$\mathcal{L}(\mathcal{D}, y) = e^{-N(y)} N(y)^{\mathcal{N}_y} \prod_{x \in \mathcal{D}} p(x|y)$$

$$t = -2 \log \frac{\mathcal{L}(\mathcal{D}, 0)}{\mathcal{L}(\mathcal{D}, 1)} = - -2 \sum_{x,y} \left[(1-y) \frac{N(0)}{\mathcal{N}_0} \left(e^{f(x)} - 1 \right) - y f(x) \right]$$