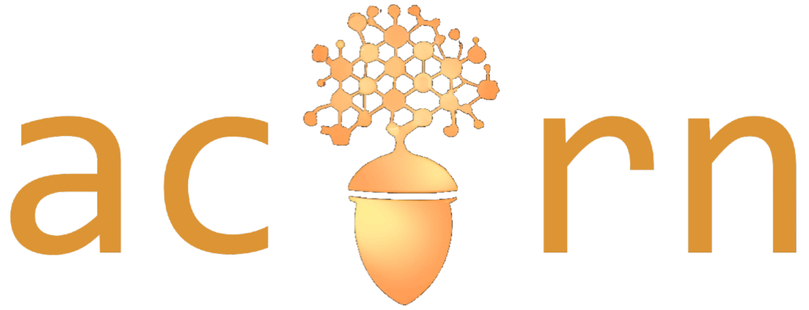
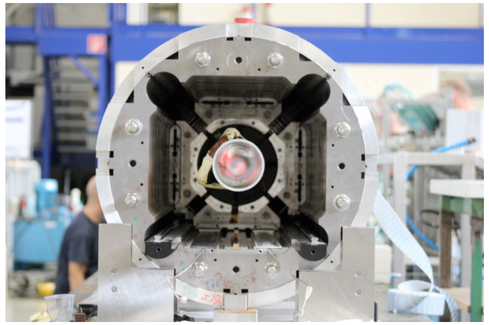
Retour d’expérience sur le développement d’un framework de R&D en IA basé sur les logiciels libres

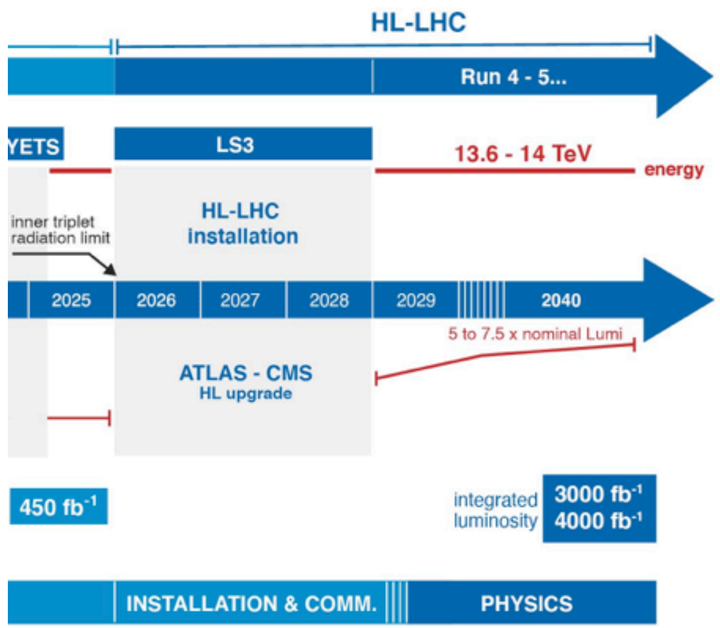
Sylvain Caillou, Laboratoire des 2 Infinis - Toulouse

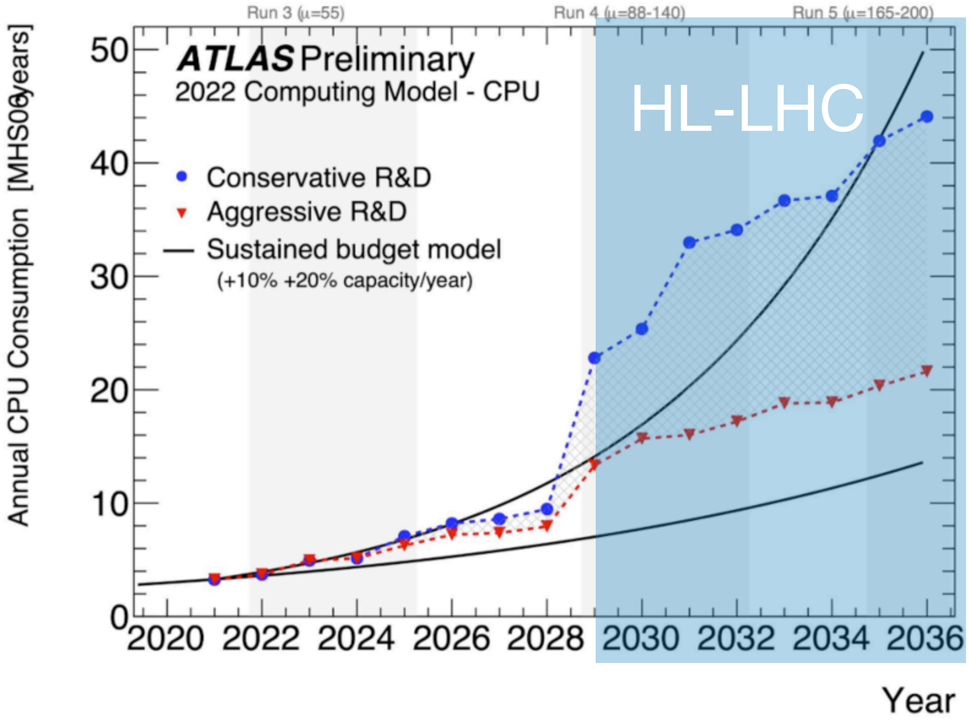


15èmes Journées Informatiques IN2P3 / IRFU, 23–26 septembre 2024

Contexte scientifique : HL-LHC







- Besoin d'une statistique plus grande en Physique des Particules
- HL-LHC (RUN 4 - 5 du LHC) doit commencer en 2029

=> Augmentation du nombre collisions (facteur 4 à 5)

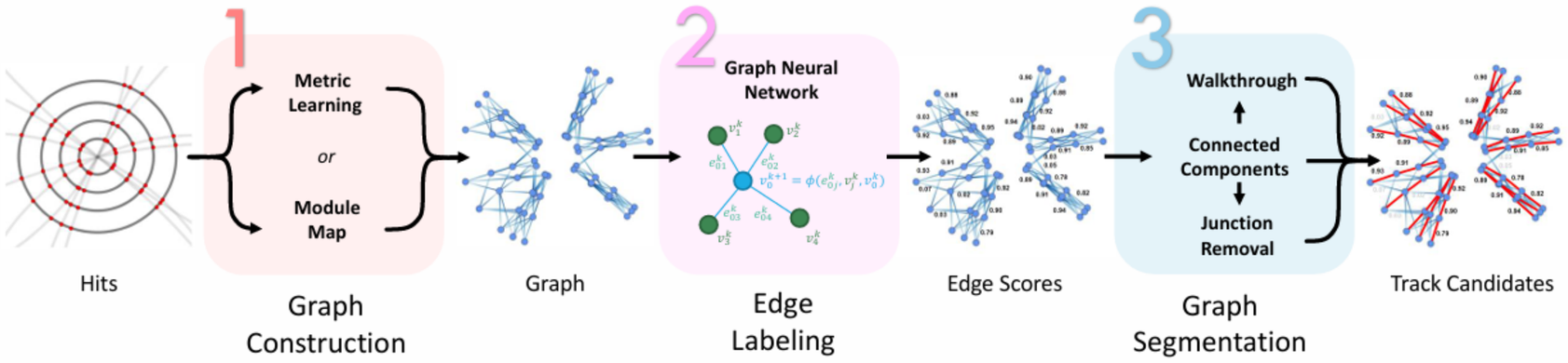
- Explosion du volume et de la complexité des données
- Phases d'upgrade de ATLAS & CMS (2026 - 2029) amélioration du hardware mais aussi du software

Un nouvel algorithme pour le tracking dans ATLAS ITK

- Run [1-3] : Combinatorial Kalmann Filter (CKF)
- Coût de calcul non linéaire avec la complexité des données
- Run [4-5] (HL-LHC) : Coût de calcul va exploser avec la combinatoire

=> Nouvelle algorithme proposé basé sur les **Graph Neural Networks (GNNs)** : projet **GNN4ITK**

=> Démonstrateur officiel d'ATLAS depuis 2023, nombreuses publications



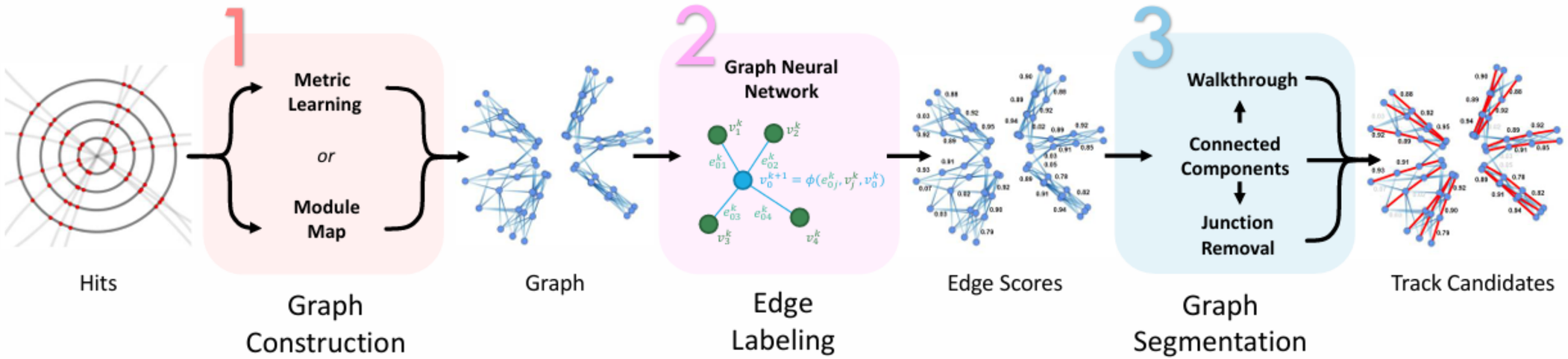
C.Rougier et al., *ATLAS ITk Track Reconstruction with a GNN-based pipeline*, Proceedings of the CTD 2022

X.Ju et al., *Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain*, Proceedings of CHEP 2023

S. Caillou et al., *Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC*, Proceedings of CHEP 2023

H. Torres et al., *Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain*, Proceedings of Connecting The Dots (CTD 2023)

GNN4ITK Pipeline



1) Représentation des données du détecteur sous forme de graphes:

Les hits sont des noeuds

Arêtes sont des connexions *possibles* entre noeuds

Les arêtes labellisées “VRAI” sont les connexions entre deux hits de la même particule

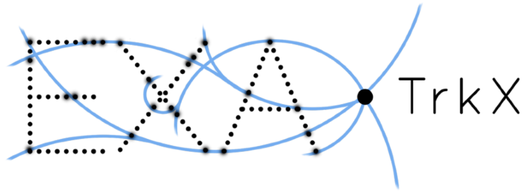
2) Un modèle GNN apprend les patterns géométriques profond des traces particules et classifie les arêtes entre VRAI et FAUSSE en donnant leur donnant un score entre 0 et 1

3) Un algorithme de segmentation opère sur les graphes avec les arêtes scorées pour construire des candidats traces

cf talk de Christophe jeudi

Le projet GNN4ITK

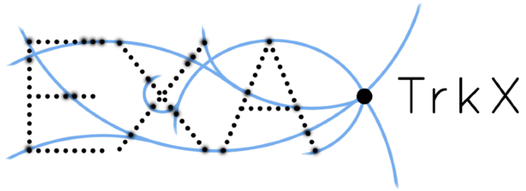
- ~35 personnes
- Collaboration étroite entre le projet Exa.TkrX (LBNL, Wisconsin university, Illinois university) et le L2IT
- Une réunion plénière hebdomadaire



Une certaine complexité :

- Données simulées par ATLAS dans les conditions HL-LHC : échantillon 100K events $t\bar{t} + \mu = \langle 200 \rangle$
- Graphes de 300K noeuds et O(1M) d'arêtes
- Des données complexes : beaucoup de features dans les données brutes
- Modèles complexes : les modèles GNNs pas si grand (~2M de paramètres) mais plein d'architectures
- De nombreux hyperparamètres : traçabilité , reproductibilité ?

Situation en 2021



BERKELEY LAB

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

ILLINOIS

L2IT

- Des pipelines parallèles dans des codes séparés
- Des codes pour chacune des étapes du pipeline dans des langages différents (C++, Python)
- Modèles GNN sous Tensorflow et PyTorch
- Des formats de données différents : définition et format de sauvegarde (csv, root)
- Définition de métriques différentes
- Outil de plot différent (matplotlib, root)

=> Comparaison des modèles ? Comparaison des résultats ? Reproductibilité ? Diffusion du code avec publication ?

Idée : Développer un framework commun

L2IT

Objectifs

Goals / Wishlist

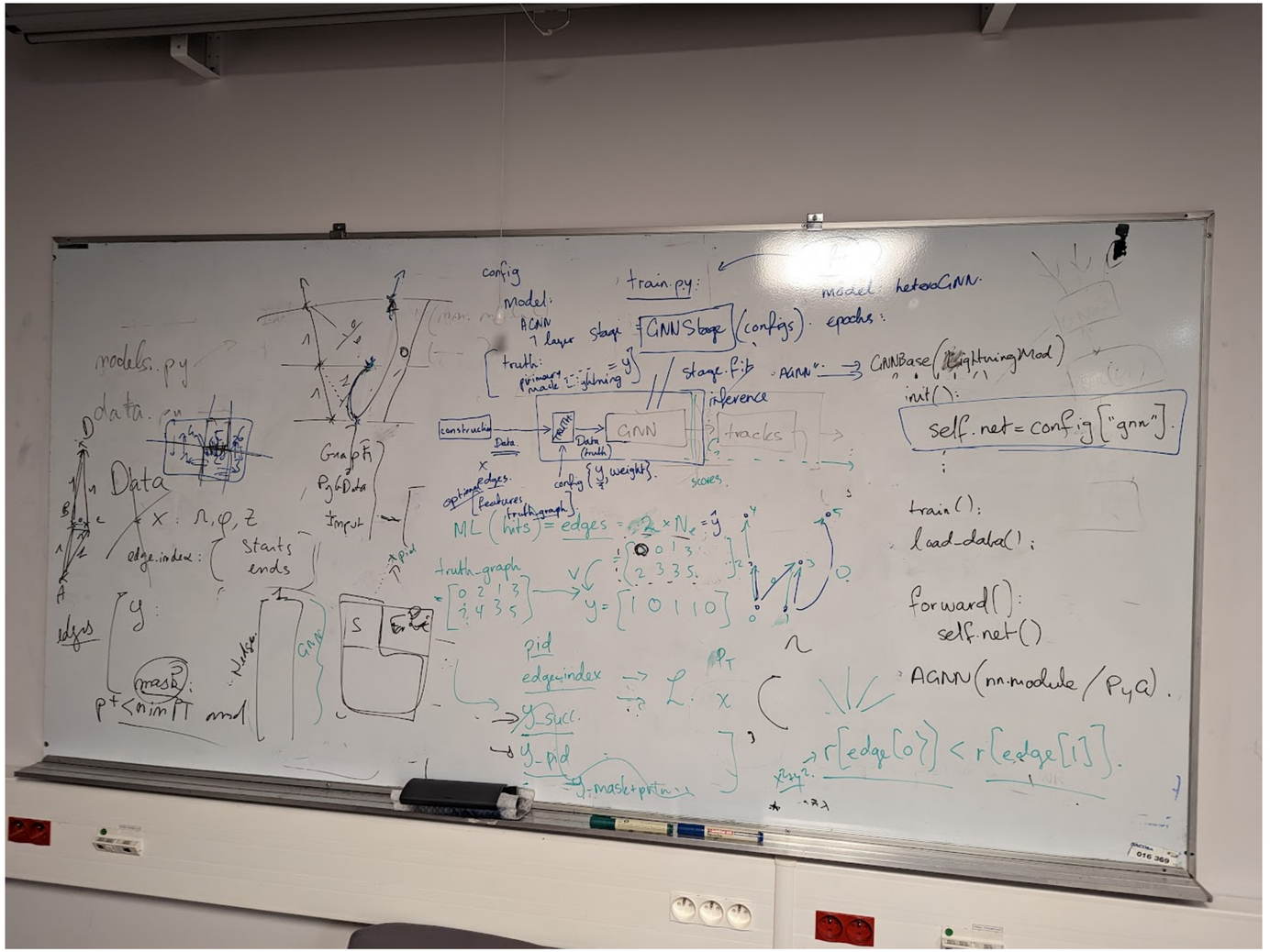
- [ ] A reference location for members of the GNN4ITk group to keep updated with each other's progress
- [ ] It should be easy to add to, easy to navigate, easy to use, easy to understand the logic
- [ ] Any member of the group should be able to easily run the pipeline training with various configurations (module map / metric learning, homo/heterogeneous GNN, iterative / recurrent)
- [ ] Any member should be able to run the pipeline in inference with various configurations - meaning there need to be pretrained models for graph construction, and models for graph edge classification (these will then need to depend on the graph construction, i.e. can choose `module_map` model for graph construction, then choose `module_map:heteroGNN` GNN model, which has been trained on module map graphs)
- [ ] Focus on python implementation for research & development
- [ ] Try to avoid use of C++ directly - if certain code is impossible/impractical to convert to python, then wrap/bind it and call it from python
- [ ] Have consistent interface - a graph can be constructed with multiple techniques, but each use the same class, the same inputs, and the same output format
- [ ] Be well-documented
- [ ] Private for the use of the GNN4ITk group *but* can freeze certain examples and configurations for public release (using a technique like this one: a public release mirror that can be cited)
- [ ] Be clean - keep messy development and random ideas out of the repository. Can include several different models and configurations, but these should have been tested already elsewhere

- Simplicité d'utilisation / User friendly
- Configurabilité / souplesse
- Généricité / Intégration de nouveaux modèles / algorithmes
- Traçabilité des expériences / Fiabilité des résultats
- Collaboration plus rapide / efficace
- Permettre la reproductibilité des résultats

Idées pour atteindre les objectifs

- Rassembler tout le pipeline dans un projet commun
- Utiliser le même langage pour tout le pipeline (Python)
- S’inspirer des meilleures des différentes expériences de développement des collaborateurs du projet
- Utiliser des logiciels et bibliothèque libre Open source pour faciliter la mise en commun et le partage
- Limiter les dépendances
- Simplicité du design pour favoriser les contributions

Conception

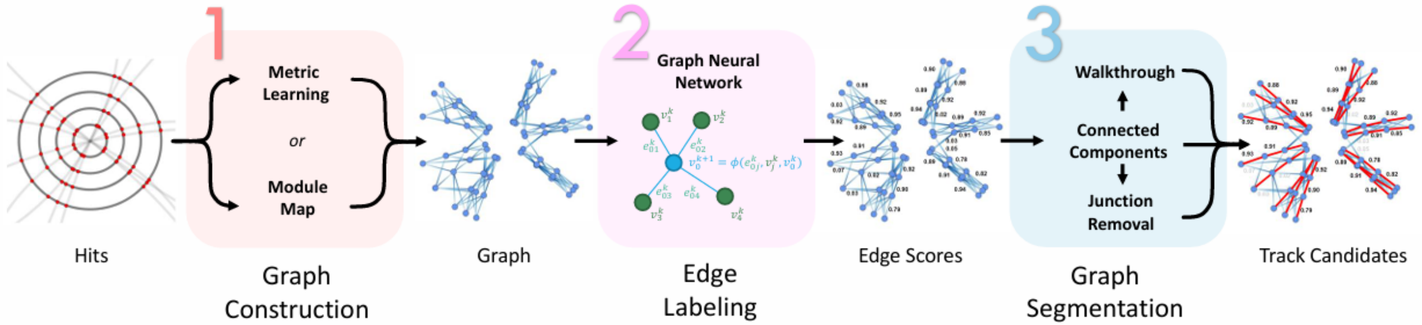


Brainstorming GNN4ITK Common Framework, Paris (LPNHE) septembre 2022

Sylvain Caillou (L2IT) & Daniel Murnane (LBNL)

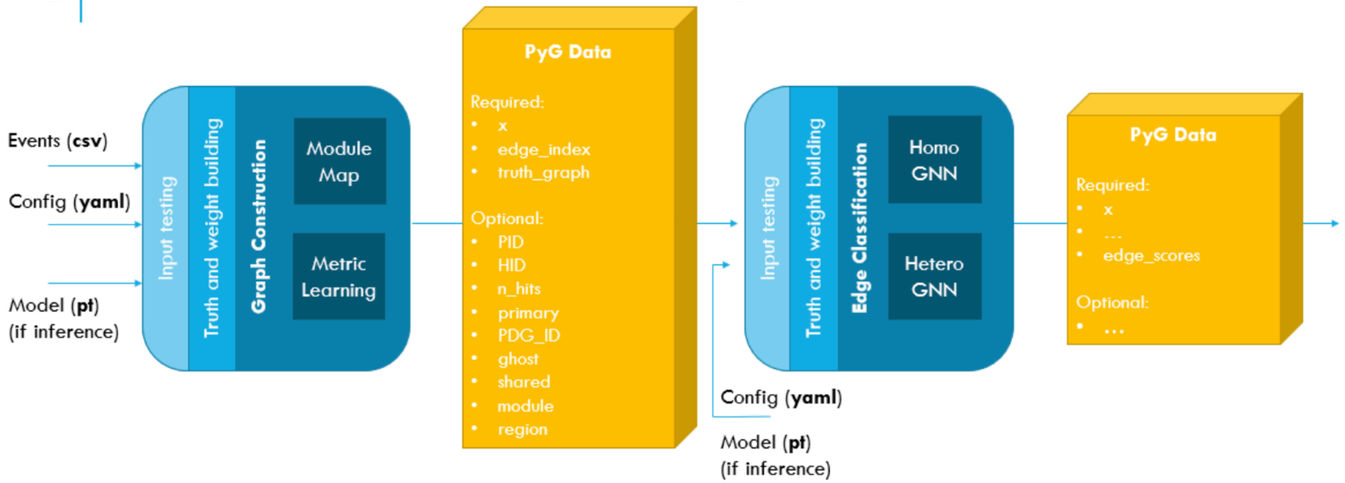
Conception

- Pour chaque étape du pipeline GNN4ITK, le code doit permettre :
  - L'entraînement des modèles
  - L'inférence des modèles
  - L'évaluation des modèles



- Design en structurant le code selon les étapes du pipeline
- Définition de métriques communes
- CLI : la plus simple possible

Implémentation, merci le libre !



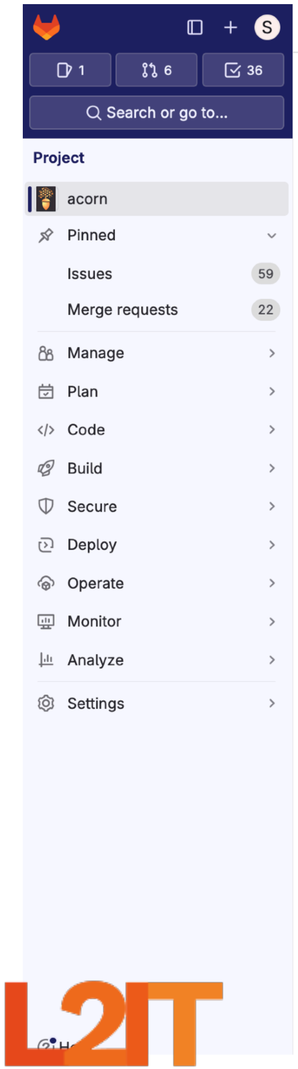
CLI simplifiée au maximum

```
acorn train $acorn_dir/examples/CTD_2023/gnn_train.yaml
acorn infer $acorn_dir/examples/CTD_2023/gnn_infer.yaml
acorn eval $acorn_dir/examples/CTD_2023/gnn_eval.yaml
```

- Représentation des graphes : PyTorch Geometric (Data object)
- Implémentation des modèles GNN : PyTorch et PyTorch Geometric
- Gestion du training PyTorch Lightning
- Preprocessing et postprocessing (plot de performances) : Numpy / Pandas / Matplotlib (ATLAS stylisé - simili ROOT)
- Configurabilité et Traçabilité : fichiers yaml pour chaque étape du pipeline + sauvegarde du contenu des fichiers yaml dans les checkpoint des modèles
- Reproductibilité : Exemple complet (README + yaml files + data + checkpoint de modèles) pour chaque publication

Et un beau jour de septembre 2022… Première publication sur le gitlab du CERN

ACORN : un framework pour l'apprentissage géométrique profond dans les détecteurs de particule



GNN4ITkTeam / acorn

README.md

LINK

documentation | pipeline passed | coverage 31.00%

This repository contains the framework used for developing, testing and presenting the GNN-based ITk track reconstruction project GNN4ITk.

Related work can be found here:

1. https://arxiv.org/abs/2103.06995
2. https://www.epj-conferences.org/articles/epjconf/abs/2021/05/epjconf_chep2021_03047/epjconf_chep2021_03047.html
3. https://cds.cern.ch/record/2815578?ln=en.

This repository is still under development and may be subject to breaking changes.

Get Started

To get started, run the setup commands (Install instructions section below), then take a look at the examples in the `examples` directory. Instructions and further details about the framework are available under the subdirectory of interest - `examples`, `acorn/stages` or `acorn/core`.

Install

IMPORTANT! Please use the `dev` branch to run all Examples: it is the latest version and is fully supported!

To install ACORN, assuming GPU capability with **cuda version >= 12.2**, run the following commands.

```
git clone --recurse-submodules ssh://git@gitlab.cern.ch:7999/gnn4itkteam/acorn.git && cd acorn
conda create --name acorn python=3.10 && conda activate acorn
pip install torch==2.1.0 && pip install --no-cache-dir -r requirements.txt
pip install -e .
```

Project information

584 Commits
107 Branches
6 Tags
232 MiB Project Storage
1 Release
1 Environment

README
Apache License 2.0
CI/CD configuration
GitLab Pages
Add CHANGELOG
Add CONTRIBUTING
Add Kubernetes cluster
Configure Integrations

Created on
September 09, 2022

- Publication publique en septembre 2023
- LICENCE.txt :
  - Apache 2 (comme le soft Athena d'ATLAS)
  - Le chef de nos collègues américain ne souhaitant pas de GPL…
  - **Déficit généralisé** de connaissances approfondies sur les licences identifié

Difficultés rencontrées

- Avant le projet:
  - Difficulté dans la reconnaissance de la nécessité d'un framework commun
  - Evolution culturel pour "sortir" le code du laboratoire et à partager les modèles / algorithmes
- Pendant le projet:
  - Demande du temps , de la communication , de la confiance commune
  - Demande des efforts pour comprendre la démarche de l'autre
  - Manque de reconnaissance et de compréhension de l'intérêt par les physiciens du temps passé à la qualité logicielle au détriment de résultats de performances de modèles plus rapide
  - Le développement collaboratif avec git et gitlab

Développement collaboratif

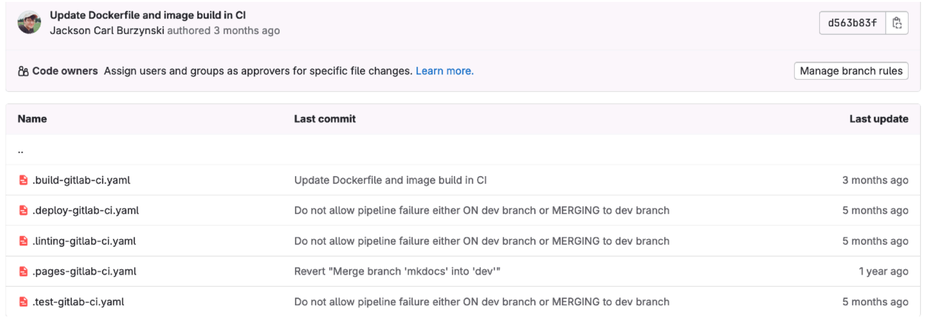
- Gestion des Merge Request : Fonctionnement distribué pas du tout efficace
- Rétropropagation des changements : S'assurer de la cohérence de la totalité du pipeline et notamment des tous les fichiers de configuration des exemples après chaque MR
- Mise en place d'un "libraire" dont le rôle est d'avoir une connaissance globale des MRs
- En pratique c'est le libraire qui passe toutes les MR après revue de code et test en local du pipeline

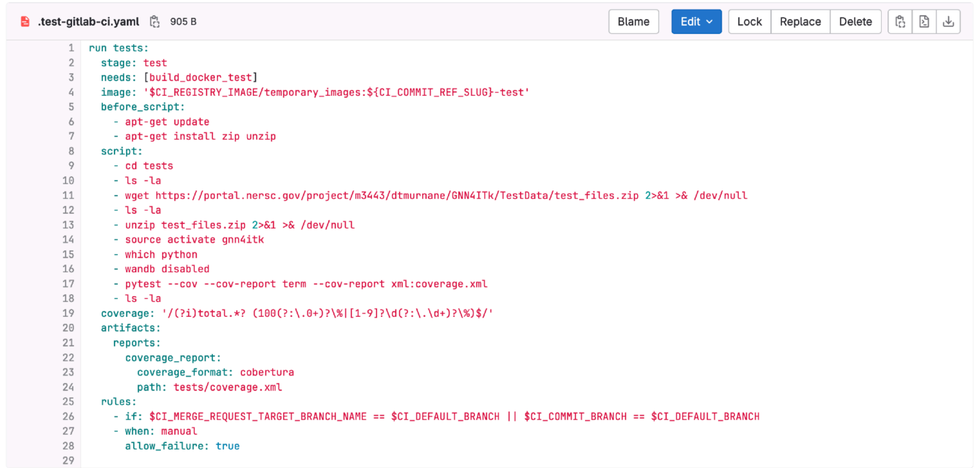
=> Extrêmement chronophage !

Développement collaboratif

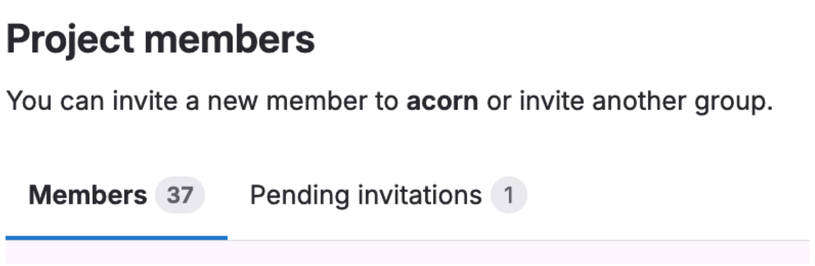
Ce qui a été mis en place:

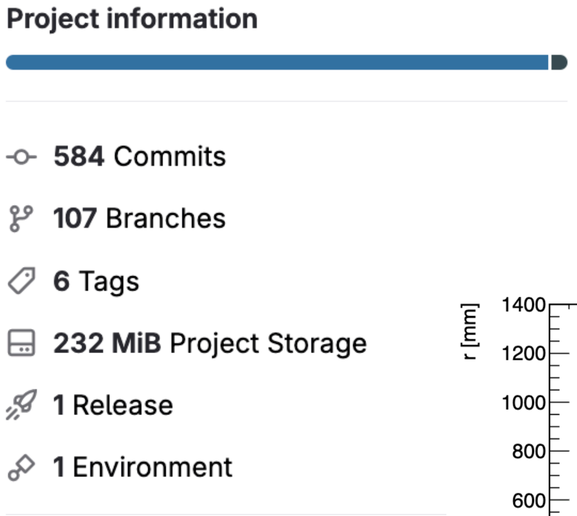
- Consignes : Les MR ne doivent concerner qu'une fonctionnalité bien identifiée et documentée (dans le message de MR) et avec peu de changements si possibles
- Des réunions dédiées GNN4ITK Dev hebdomadaires en plus des réunions plénières pour passer en revue MRs et Issues
- Un canal slack GNN4ITK Dev dédié
- Un pipeline CI:
  - linter + tests unitaires
  - Mauvais coverage (~30%, doit être amélioré)
  - Nécessité de tests fonctionnels sur les résultats (i.e. analyse de distribution)

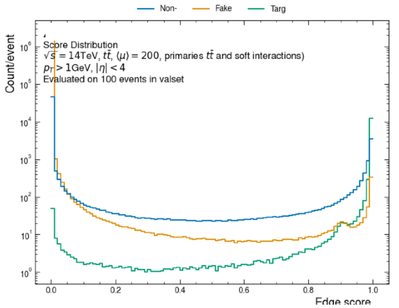


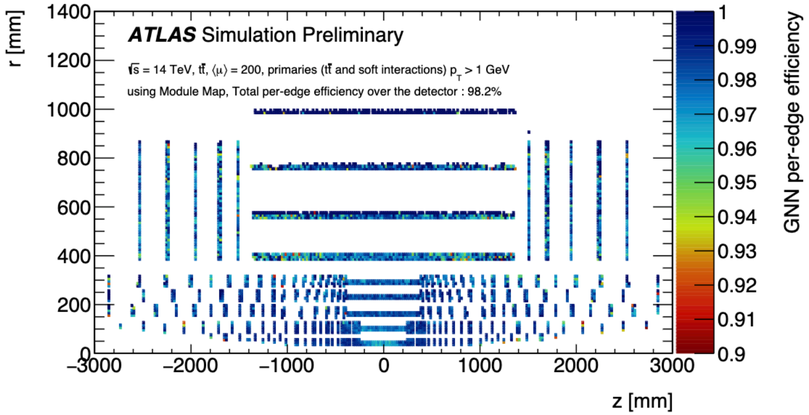


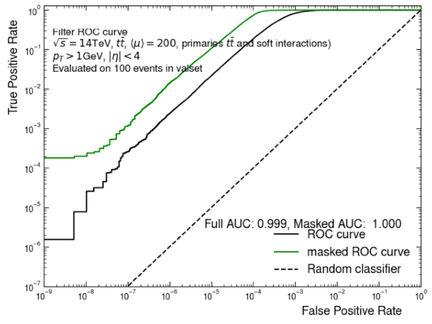
Où en est le projet au bout de 2 ans ?





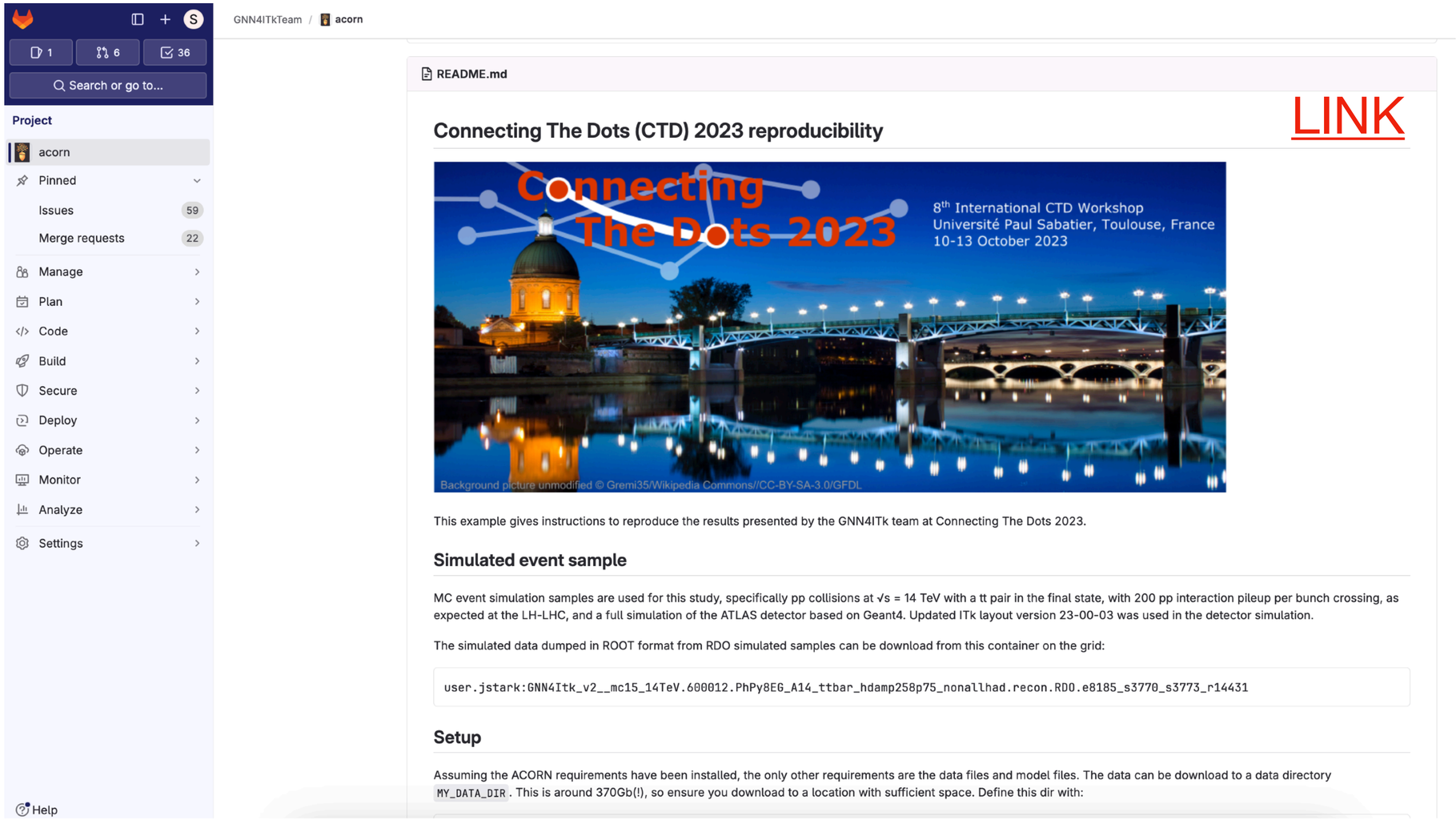






- ~40 membres
- 15 post-docs et doctorants utilisateurs intensifs
- 7 contributeurs principaux (dont des doctorants des post-doc des chercheurs et ingénieurs)
- Acorn est l'unique référence pour la R&D dans GNN4ITK: fournit tous les plots de performances
- Interface avec IDPVM pour les performances de physique
- En cours d'interfaçage avec ACTS et Athena pour inférence des modèles (vers le déploiement en production)

Reproductibilité: exemple de CTD23



H. Torres et al., *Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain*, Proceedings of Connecting The Dots (CTD 2023)

Quelques dernières réflexions ….

- Inférence pipeline portée sur ACTS (B. Hugues et C. Collard) et sur Athena (X. Ju): L'utilisation du libre peut aussi poser quelques dépendances / limites. Exemples:
  - torch.compile qui optimise l'inférence en Python mais n'est pas clairement porté sous libtorch (C++)
  - PyTorch Geometric n'est pas nativement implémentée en C++ : difficultés d'interface entre Acorn et ACTS

Et perspectives pour la suite….

- Zenodo (pour un meilleur partage du code et des données)
- Methodologies pour l'ingénierie logicielle en R&D (cf talk de Philip hier :-))
- Test de résultats de distribution des résultats -> A developper pour un pipeline CI beaucoup plus efficace !
- IA utile ? RAGs -> systèmes pour guider l'ingénierie logicielle en R&D ?

Merci de votre attention 🙏