
GNN pour la reconstruction de traces sur LHCb / RTA

Anthony Correia, Fotis Giasemis, Nabil Garroum,
Vava Gligorov

20/11/2023

Allen / RTA / Real Time Analysis

RTA / LHCb:

Allen est le déclencheur de l'expérience LHCb, basé sur la technologie GPU (Cluster de ~500 GPU)
Qualifié en 2022, en cours d'utilisation par la collaboration. Performance meilleurs qu'attendu.

Structure Allen utilisée actuellement dans le stack LHCb

Standalone (Sans le Stack LHCb)

Intégration dans le stack : 2 couches MPI

- MPI géré par Moore
- MPI géré par Allen : répartition des paquets d'événements (~15) sur les GPUs

Evolution possible vers le deep-learning grâce aux GPUs

DD4HEP : (Detector Description for high Energy Physics): Un formalisme de description des détecteur incluant la reconstruction, la simulation et l'analyse.

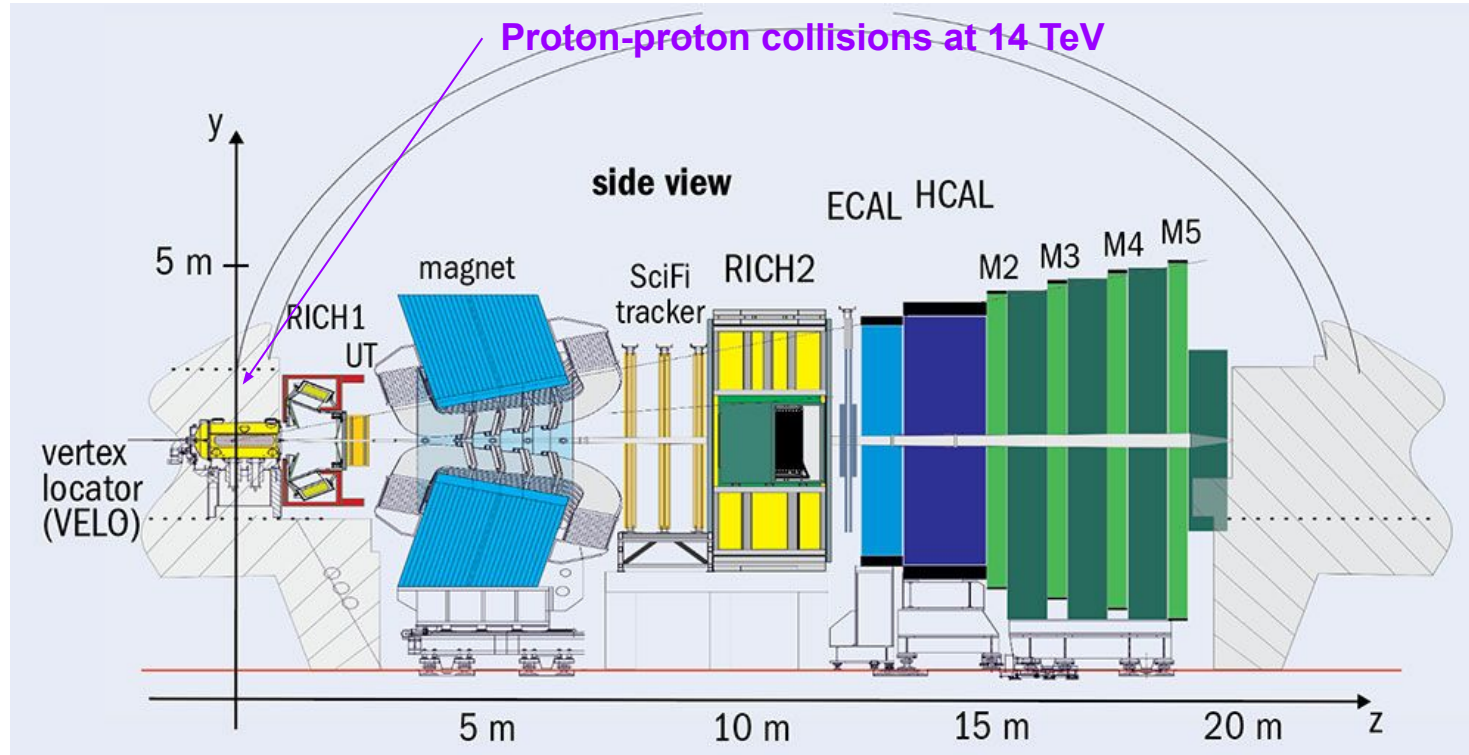
Intégration de DD4HEP dans Allen en 2022.

Version en cours d'utilisation. Mais Quelques MRs

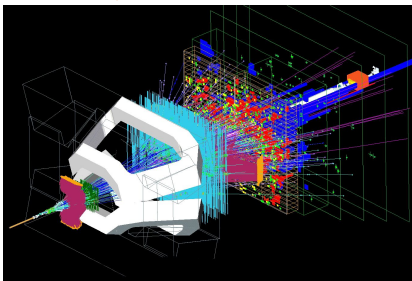
Allen : C++ / Cuda / Python / Cmake ,
CI/CD Forge Gitlab : ~100000 lignes
Version Standalone ou LHCb Stack
2200 hits / Collision , ~150 traces ou
particules

Détecteur LHCb

Détecteur LHCb après “upgrade” (2022 - ...)



RTA (Real Time Analysis) / Allen



Collisions (Run 3)

- 20 MHz non-empty bunch crossing rate
- ~ 5 collisions / bunch crossing
- $p-p$ collision at $\sqrt{s} = 13.6$ TeV

LHCb Subdetectors

5 TB/s

Allen (High-Level Trigger 1)
fully GPU-based online partial reconstruction and selection

70-200 GB/s

Storage buffer

Geometry

Better trigger efficiency than previous L0 FPGA-based trigger

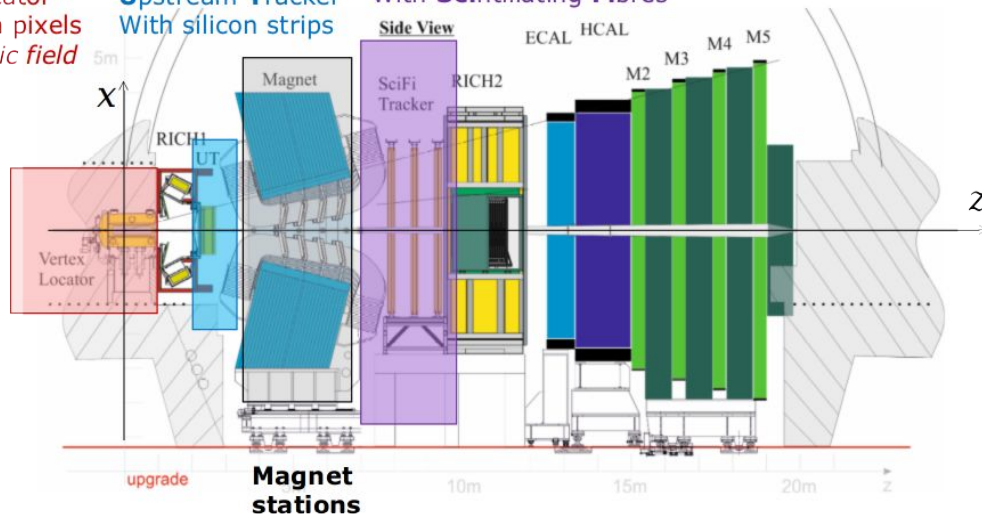
High-Level Trigger 2
CPU-based full reconstruction and selection

10 GB/s

Velo
Vertex Locator
With silicon pixels
No magnetic field

UT
Upstream Tracker
With silicon strips

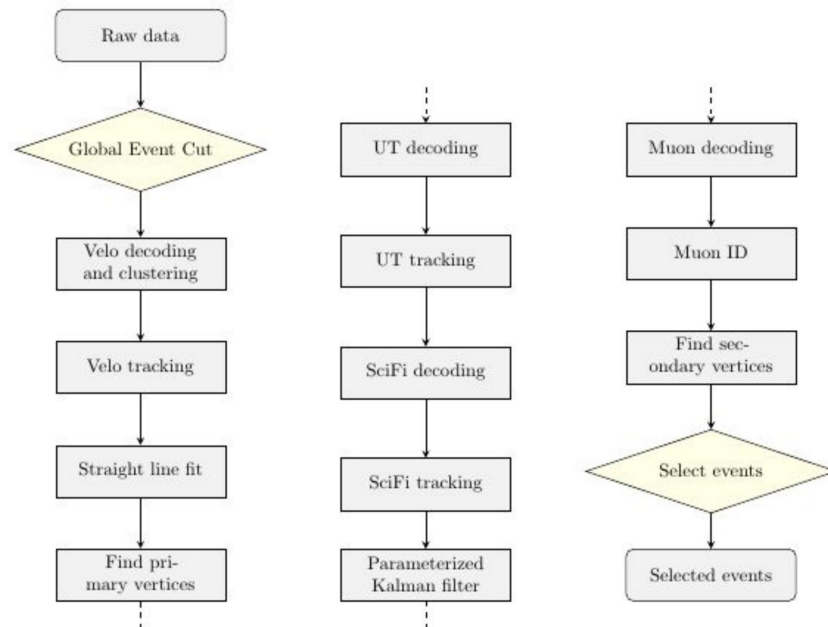
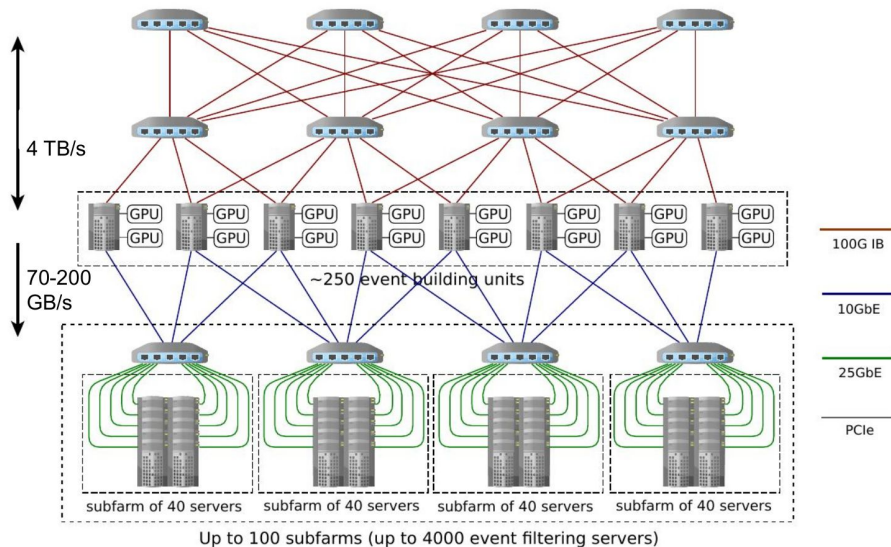
SciFi
With Scintillating Fibres



Numbers taken from [LHCb-FIGURE-2020-016](#)

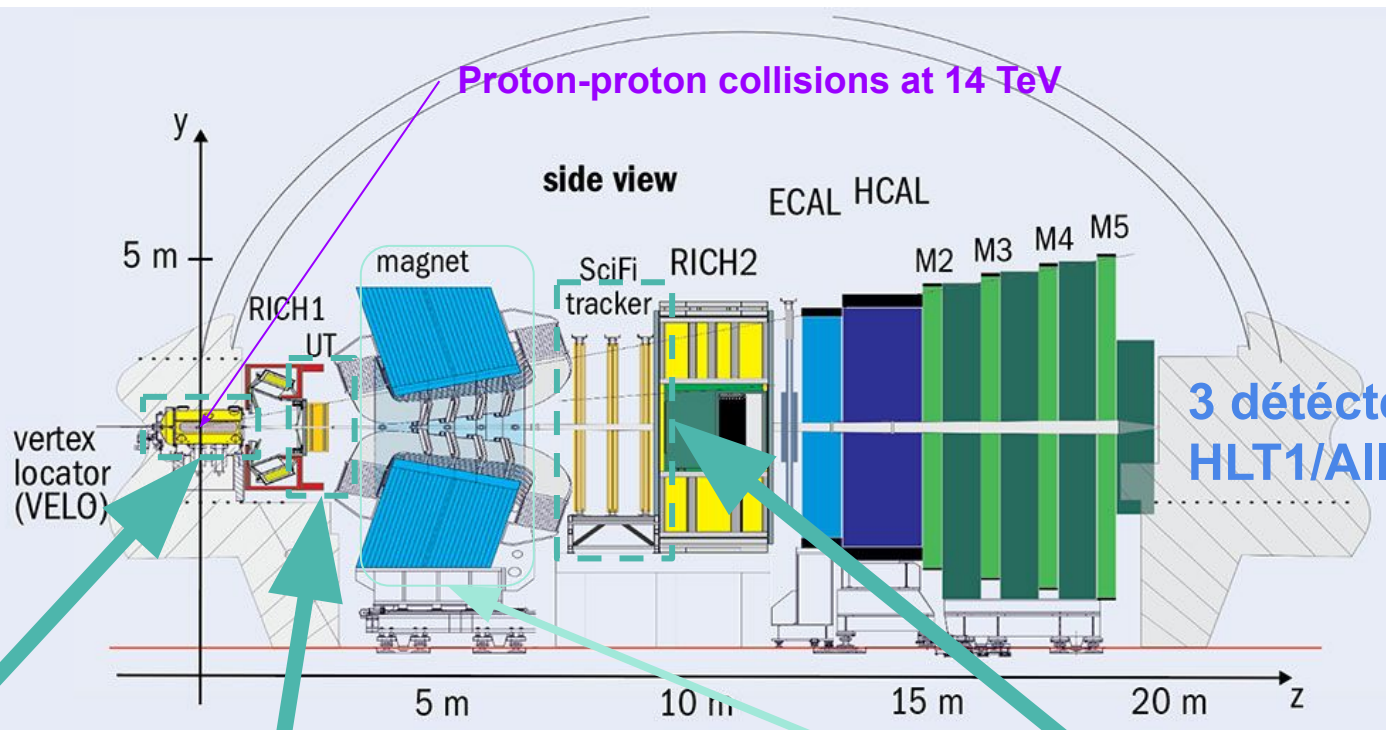
RTA / Allen

Séquence d'algorithmes exécutés sur les GPUs (10 events / GPU)



RTA (Real Time Analysis) / Allen

LHCb detector after “upgrade Ia” (2022 - ...)



1 Velo
(Vertex Locator)

2 UT
(Upstream Tracker)

Magnet

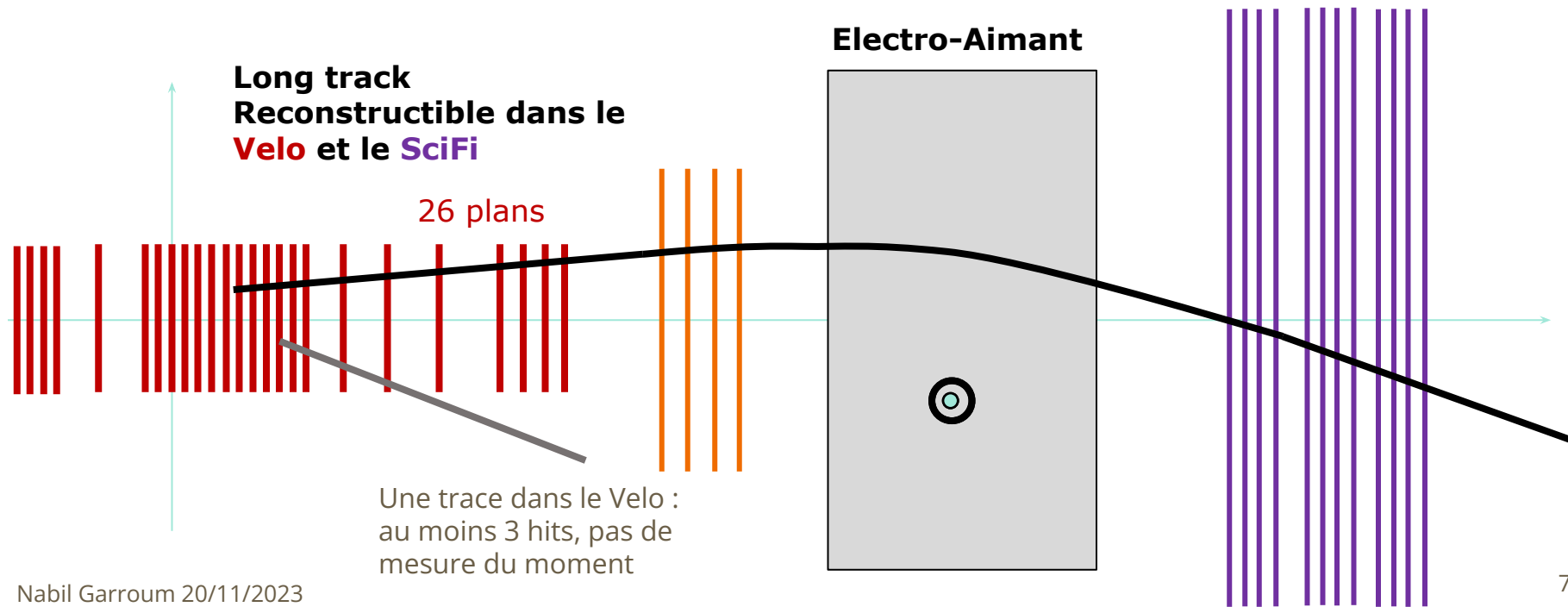
3 SciFi
(Scintillating fiber tracker)

RTA (Real Time Analysis) / HLT1/Allen

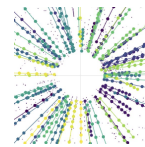
Velo
Vertex **L**ocator
With silicon pixels

UT
Upstream **T**racker
With silicon strips

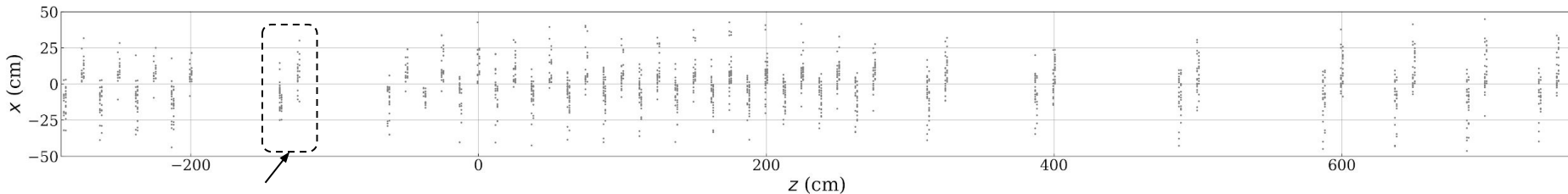
SciFi
With **S**ci^{tillating} **F**ibres



Reconstruction de traces sur le Velo

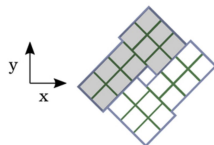


Objectif : Reconstruction de traces

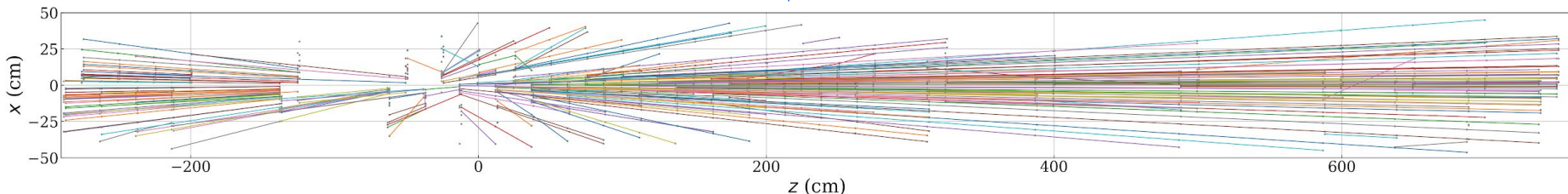


26 plans

1 plan \equiv 4 modules

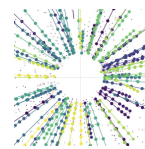


Reconstruction de traces : linéaires
ou linéaire par morceaux



Notre approche: reconstruction de traces basée sur le **Deep Learning**

Reconstruction de traces sur le Velo par GNN



Travaux précédents

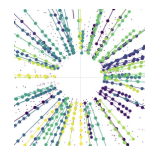
- [ML tracking avec OneAPI à l'Université de Liverpool](#): reconstruction de traces dans le Velo, basé sur le machine learning , implémenté sur FPGA
- Collaboration [Exa.TrkX](#)
 - [Pipeline de Deep-Learning basé sur les Graph Neural Networks](#) (GNN)
 - Originellement pour ATLAS [TrackML](#) dataset, mais également testé pour [DUNE TPC](#) et [CMS HGCAL](#)
 - Ont généreusement partagé la pipeline dans un répo dédié

Motivations

- [Exa.TrkX](#) a démontré une inférence linéaire en fonction en # de hits , mais non testé sur les **structures en Jet**
- Algorithmes actuels, au mieux quadratiques.
- Futurs upgrades d'HLCb: Augmentation de la luminosité et de la granularité, nécessité d'un trigger plus rapide

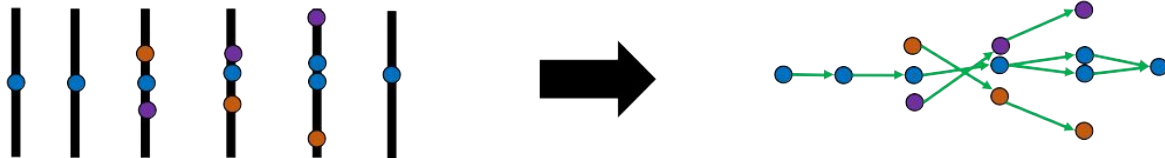
Framework Allen : implémentation des algorithmes combinatoire et le GNN sur une même base pour une comparaison pertinentes : **Efficacité et Inférence**

Flux de travail



Graphe : outil intuitif et naturel pour la reconstruction de traces

GNN : généralisation des réseaux de neurones, moins de limitations sur les dimensions d'entrée et de sortie



Dans un premier temps : Séparer les sous-détecteurs
Comparer l'efficacité et l'inférence pour le Velo

GDL4HEP , forge Gitlab du CERN : Apprentissage sur les GPUs du LIP6 (A100).

Pytorch / mlFlow (MLops) pour le suivi des modèles



Utilisation des simulations de la grille
[XDIGI2CSV repo.](#)

[etx4velo](#), fork depuis Exa.TrkX.

[MonteTracko library.](#)
Implémenté uniquement pour le Velo.

Déploiement des modèles PyTorch dans Allen pour comparer la rapidité de reconstruction

DataSet pour l'apprentissage : Xdigi2csv



⇒ Utilisation des événements minimum-bias sur la grille du CERN format .xdigi

- **Simulations réalistes**
- La performance de reconstruction peut être comparée à Allen

⇒ Fichiers .parquet format, avec compression lz4

- **Facile à lire en Python** → pour une utilisation avec PyTorch
- **Rapide en lecture/écriture** → traitement rapide des datas
- **Léger** → Sauvegarde / téléchargement > **1 500 000 events**

⇒ **XDIGI2CSV repository** ([repository](#), [documentation](#))

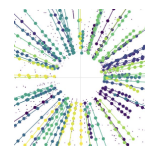
- Soumission .xdigi → **.parquet.lz4 conversion** à HTCondor batch system depuis un **simple fichier YAML**
- **Data Frame** [peut être téléchargé par toute la collaboration](#)
- > **2 million événements disponibles** (236 GB)

hits_velo.parquet.lz4

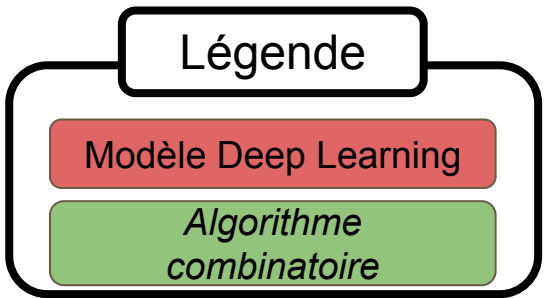
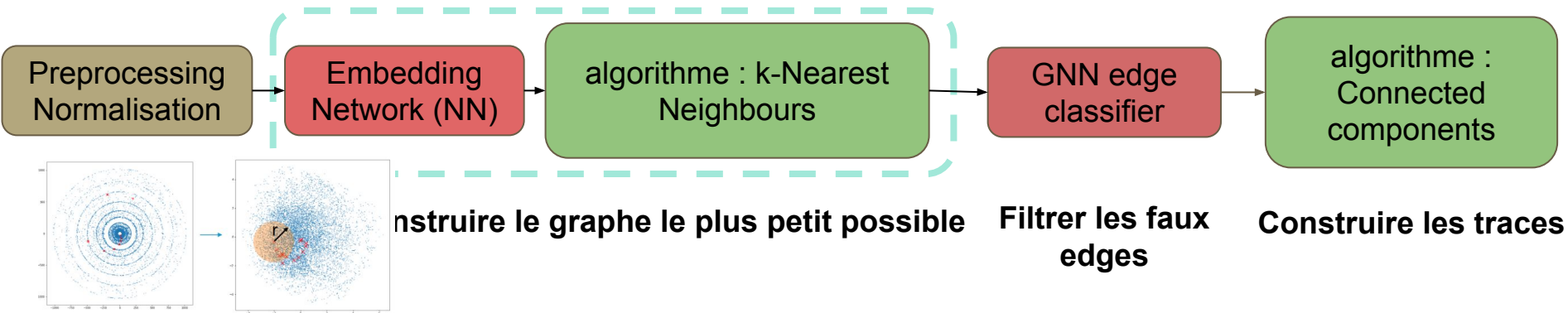
event ID	hit ID	particle ID	x, y, z, plane, ...
----------	--------	-------------	---------------------

mc_particles.parquet.lz4

event ID	particle ID	pid, pt, eta, phi, ...
----------	-------------	------------------------



ETX4VELO: Pipeline (python)



Pour l'inférence : Les modèles de deep learning doivent être exporté vers Allen, la pipeline combine le deep learning et la combinatoire (intrinsèque à la construction du graphe)

Merci Fotis...

ETX4VELO: Pipeline (python) : NN

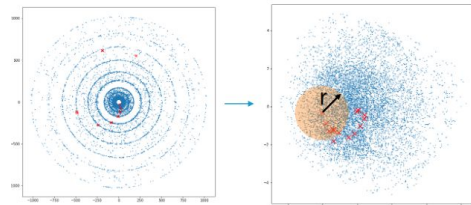
1. Construire l'espace latent

Cylindrical coordinates

$(\vec{r}, \phi, z, \text{plane})$

Dense Neural Network
(DNN)
35K parameters

$\vec{e} = (e_1, e_2, e_3, e_4)$



Paralléliser sur les hits

Si deux hits forment un edge

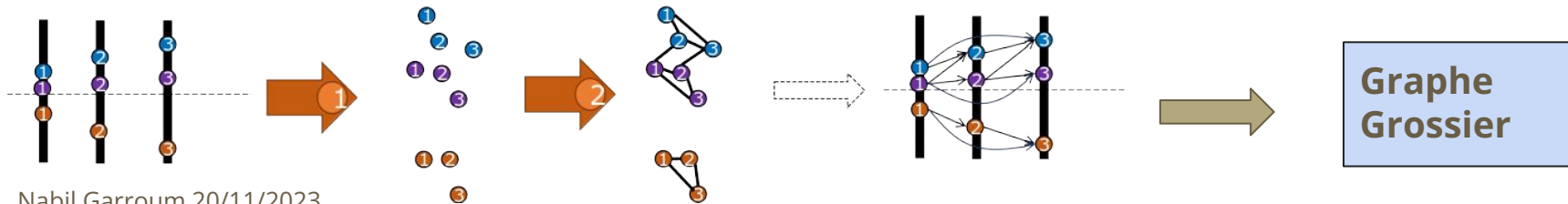
$$d(A, B)^2 = \|\vec{e}_A - \vec{e}_B\|^2 < 0.010$$

Sinon

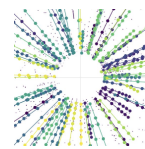
$$d(A, B)^2 > 0.010$$

2. Boucler sur le plan : Appliquer kNN (plus proche voisin) entre les hits et les plans p, p+1 et p+2

Paralléliser sur les hits

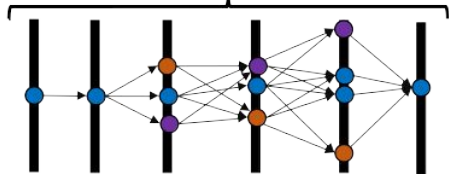


ETX4VELO: Inference Pipeline (python) : Graphe-Edges

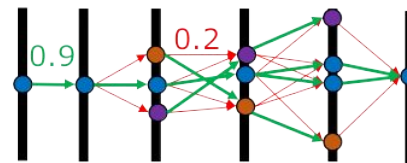


Graphe basé sur les edges (connexions entre hits)

Output of Embedding + kNN

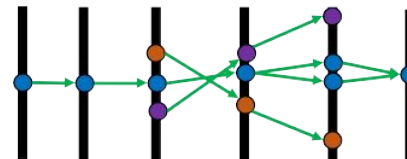


GNN edge classifier
score for every edge



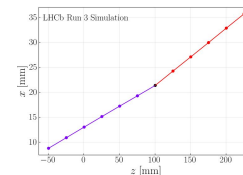
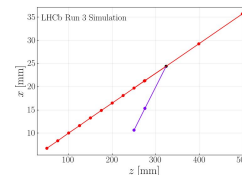
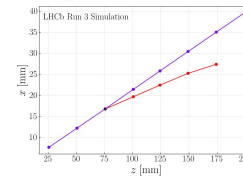
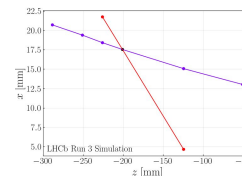
Edge score cut

>



Metric	Allen	etx4velo
Efficiency	98.17%	46.23%
Clone rate	3.07%	0.47%
Hit efficiency	95.35%	98.89%
Hit purity	99,67%	93.89%

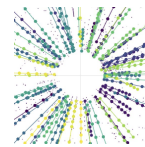
La reconstruction pour les les long-électrons est mauvaise : les hits en commun sont difficiles à classer



~ 55 % des électrons partagent un hits avec un autre électron

Évalué sur 5000 événements

ETX4VELO: Pipeline : Graphe-Triplets



Graphe basé sur les triplets (connexions entre edges)

Classe les hits partagés en 3 catégories, cela couvre toutes les cas possibles :

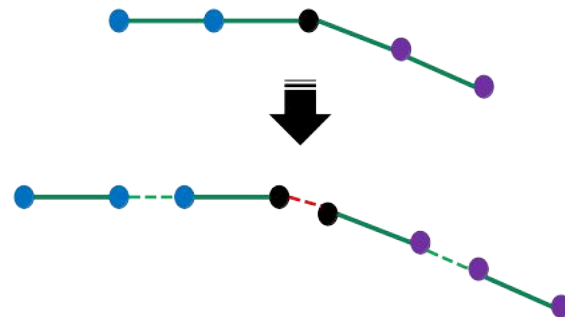
Articulation



Left elbow



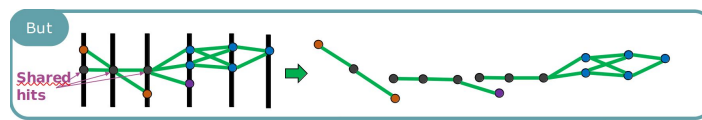
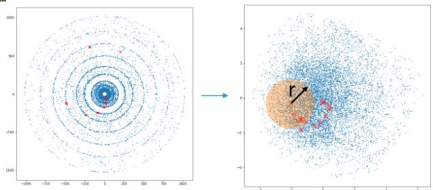
Right elbow



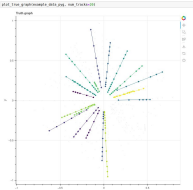
Merci Anthony...

Utilisation d'un GNN basé sur les triplets, résultats

DataSet
X, Y, Z
R, Phi, Z



Hits Partagés



```

jupyter pipeline_config.yaml 03/07/2023
File Edit View Language

1 common_configs:
2 experiment_name: velo-smbias-sih00-xdigi
3 artifact_directory: artifacts
4 gpus: 1
5 clear_directories: True
6
7 processing_configs:
8 input_dir: data/input
9 preprocessed_dir: data/preprocessed
10 output_dir: data/processed
11 n_files: 100
12 generate: True
13 noise: True
14 n_workers: 32
15 feature_size: [50, 3, 14159, 200] # For Velo
16 needs_preprocessing: True # True for Velo, False for
17 non_true_hits_threshold: 0
18
19 metric_learning_configs:
20 # Dataset parameters
21 input_dir: data/preprocessed
22 output_dir: data/metric_learning_processed
23 pt_signal_cut: .08
24 pt_background_cut: .08
25 train_split: [80, 10, 10] # Number of [training, val,
26 true_edges: modewise_true_edges
27
28 # Model parameters
29 n_channels: 3
30 cell_channels: 0
31 nb_hitters: 250
32 nb_layers: 4
33 emb_dim: 3
34 activation: Tanh
35 weight: 2
36 randomization: 2
37 points_per_batch: 100000
38 r_train: 0.01 # approximately 1/20 of the scale
39 r_val: 0.01
40 r_test: 0.01
41 bias: 50
42 margin: 0
43 wmargin: 0
44 lr: 0.001
45 factor: 0.7
46 regime: 4
47 regime: [rp, hm, norm]
48 max_epochs: 20
49
50 -----
51 @): Adjust size = 3
52 device: [0, 1]
53
54 Factor advice
55
56 @): def make_kgpt
57 class KGPT
58 """
59 Metric_learning_configs:
60 metric_activation: 'ReLU'
61 metric_activation: 'ReLU'
62 metric_activation: 'ReLU'
63 loss: 'triplet_loss'
64
65 """
66 @): @staticmethod as MLP-MLP specified fully connected layers, ...
67 """
68 """
69 """
70 """
71 """
72 """
73 """
74 """
75 """
76 """
77 """
78 """
79 """
80 """
81 """
82 """
83 """
84 """
85 """
86 """
87 """
88 """
89 """
90 """
91 """
92 """
93 """
94 """
95 """
96 """
97 """
98 """
99 """
100 """
101 """
102 """
103 """
104 """
105 """
106 """
107 """
108 """
109 """
110 """
111 """
112 """
113 """
114 """
115 """
116 """
117 """
118 """
119 """
120 """
121 """
122 """
123 """
124 """
125 """
126 """
127 """
128 """
129 """
130 """
131 """
132 """
133 """
134 """
135 """
136 """
137 """
138 """
139 """
140 """
141 """
142 """
143 """
144 """
145 """
146 """
147 """
148 """
149 """
150 """
151 """
152 """
153 """
154 """
155 """
156 """
157 """
158 """
159 """
160 """
161 """
162 """
163 """
164 """
165 """
166 """
167 """
168 """
169 """
170 """
171 """
172 """
173 """
174 """
175 """
176 """
177 """
178 """
179 """
180 """
181 """
182 """
183 """
184 """
185 """
186 """
187 """
188 """
189 """
190 """
191 """
192 """
193 """
194 """
195 """
196 """
197 """
198 """
199 """
200 """
201 """
202 """
203 """
204 """
205 """
206 """
207 """
208 """
209 """
210 """
211 """
212 """
213 """
214 """
215 """
216 """
217 """
218 """
219 """
220 """
221 """
222 """
223 """
224 """
225 """
226 """
227 """
228 """
229 """
230 """
231 """
232 """
233 """
234 """
235 """
236 """
237 """
238 """
239 """
240 """
241 """
242 """
243 """
244 """
245 """
246 """
247 """
248 """
249 """
250 """
251 """
252 """
253 """
254 """
255 """
256 """
257 """
258 """
259 """
260 """
261 """
262 """
263 """
264 """
265 """
266 """
267 """
268 """
269 """
270 """
271 """
272 """
273 """
274 """
275 """
276 """
277 """
278 """
279 """
280 """
281 """
282 """
283 """
284 """
285 """
286 """
287 """
288 """
289 """
290 """
291 """
292 """
293 """
294 """
295 """
296 """
297 """
298 """
299 """
300 """
301 """
302 """
303 """
304 """
305 """
306 """
307 """
308 """
309 """
310 """
311 """
312 """
313 """
314 """
315 """
316 """
317 """
318 """
319 """
320 """
321 """
322 """
323 """
324 """
325 """
326 """
327 """
328 """
329 """
330 """
331 """
332 """
333 """
334 """
335 """
336 """
337 """
338 """
339 """
340 """
341 """
342 """
343 """
344 """
345 """
346 """
347 """
348 """
349 """
350 """
351 """
352 """
353 """
354 """
355 """
356 """
357 """
358 """
359 """
360 """
361 """
362 """
363 """
364 """
365 """
366 """
367 """
368 """
369 """
370 """
371 """
372 """
373 """
374 """
375 """
376 """
377 """
378 """
379 """
380 """
381 """
382 """
383 """
384 """
385 """
386 """
387 """
388 """
389 """
390 """
391 """
392 """
393 """
394 """
395 """
396 """
397 """
398 """
399 """
400 """
401 """
402 """
403 """
404 """
405 """
406 """
407 """
408 """
409 """
410 """
411 """
412 """
413 """
414 """
415 """
416 """
417 """
418 """
419 """
420 """
421 """
422 """
423 """
424 """
425 """
426 """
427 """
428 """
429 """
430 """
431 """
432 """
433 """
434 """
435 """
436 """
437 """
438 """
439 """
440 """
441 """
442 """
443 """
444 """
445 """
446 """
447 """
448 """
449 """
450 """
451 """
452 """
453 """
454 """
455 """
456 """
457 """
458 """
459 """
460 """
461 """
462 """
463 """
464 """
465 """
466 """
467 """
468 """
469 """
470 """
471 """
472 """
473 """
474 """
475 """
476 """
477 """
478 """
479 """
480 """
481 """
482 """
483 """
484 """
485 """
486 """
487 """
488 """
489 """
490 """
491 """
492 """
493 """
494 """
495 """
496 """
497 """
498 """
499 """
500 """
501 """
502 """
503 """
504 """
505 """
506 """
507 """
508 """
509 """
510 """
511 """
512 """
513 """
514 """
515 """
516 """
517 """
518 """
519 """
520 """
521 """
522 """
523 """
524 """
525 """
526 """
527 """
528 """
529 """
530 """
531 """
532 """
533 """
534 """
535 """
536 """
537 """
538 """
539 """
540 """
541 """
542 """
543 """
544 """
545 """
546 """
547 """
548 """
549 """
550 """
551 """
552 """
553 """
554 """
555 """
556 """
557 """
558 """
559 """
560 """
561 """
562 """
563 """
564 """
565 """
566 """
567 """
568 """
569 """
570 """
571 """
572 """
573 """
574 """
575 """
576 """
577 """
578 """
579 """
580 """
581 """
582 """
583 """
584 """
585 """
586 """
587 """
588 """
589 """
590 """
591 """
592 """
593 """
594 """
595 """
596 """
597 """
598 """
599 """
600 """
601 """
602 """
603 """
604 """
605 """
606 """
607 """
608 """
609 """
610 """
611 """
612 """
613 """
614 """
615 """
616 """
617 """
618 """
619 """
620 """
621 """
622 """
623 """
624 """
625 """
626 """
627 """
628 """
629 """
630 """
631 """
632 """
633 """
634 """
635 """
636 """
637 """
638 """
639 """
640 """
641 """
642 """
643 """
644 """
645 """
646 """
647 """
648 """
649 """
650 """
651 """
652 """
653 """
654 """
655 """
656 """
657 """
658 """
659 """
660 """
661 """
662 """
663 """
664 """
665 """
666 """
667 """
668 """
669 """
670 """
671 """
672 """
673 """
674 """
675 """
676 """
677 """
678 """
679 """
680 """
681 """
682 """
683 """
684 """
685 """
686 """
687 """
688 """
689 """
690 """
691 """
692 """
693 """
694 """
695 """
696 """
697 """
698 """
699 """
700 """
701 """
702 """
703 """
704 """
705 """
706 """
707 """
708 """
709 """
710 """
711 """
712 """
713 """
714 """
715 """
716 """
717 """
718 """
719 """
720 """
721 """
722 """
723 """
724 """
725 """
726 """
727 """
728 """
729 """
730 """
731 """
732 """
733 """
734 """
735 """
736 """
737 """
738 """
739 """
740 """
741 """
742 """
743 """
744 """
745 """
746 """
747 """
748 """
749 """
750 """
751 """
752 """
753 """
754 """
755 """
756 """
757 """
758 """
759 """
760 """
761 """
762 """
763 """
764 """
765 """
766 """
767 """
768 """
769 """
770 """
771 """
772 """
773 """
774 """
775 """
776 """
777 """
778 """
779 """
780 """
781 """
782 """
783 """
784 """
785 """
786 """
787 """
788 """
789 """
790 """
791 """
792 """
793 """
794 """
795 """
796 """
797 """
798 """
799 """
800 """
801 """
802 """
803 """
804 """
805 """
806 """
807 """
808 """
809 """
810 """
811 """
812 """
813 """
814 """
815 """
816 """
817 """
818 """
819 """
820 """
821 """
822 """
823 """
824 """
825 """
826 """
827 """
828 """
829 """
830 """
831 """
832 """
833 """
834 """
835 """
836 """
837 """
838 """
839 """
840 """
841 """
842 """
843 """
844 """
845 """
846 """
847 """
848 """
849 """
850 """
851 """
852 """
853 """
854 """
855 """
856 """
857 """
858 """
859 """
860 """
861 """
862 """
863 """
864 """
865 """
866 """
867 """
868 """
869 """
870 """
871 """
872 """
873 """
874 """
875 """
876 """
877 """
878 """
879 """
880 """
881 """
882 """
883 """
884 """
885 """
886 """
887 """
888 """
889 """
890 """
891 """
892 """
893 """
894 """
895 """
896 """
897 """
898 """
899 """
900 """
901 """
902 """
903 """
904 """
905 """
906 """
907 """
908 """
909 """
910 """
911 """
912 """
913 """
914 """
915 """
916 """
917 """
918 """
919 """
920 """
921 """
922 """
923 """
924 """
925 """
926 """
927 """
928 """
929 """
930 """
931 """
932 """
933 """
934 """
935 """
936 """
937 """
938 """
939 """
940 """
941 """
942 """
943 """
944 """
945 """
946 """
947 """
948 """
949 """
950 """
951 """
952 """
953 """
954 """
955 """
956 """
957 """
958 """
959 """
960 """
961 """
962 """
963 """
964 """
965 """
966 """
967 """
968 """
969 """
970 """
971 """
972 """
973 """
974 """
975 """
976 """
977 """
978 """
979 """
980 """
981 """
982 """
983 """
984 """
985 """
986 """
987 """
988 """
989 """
990 """
991 """
992 """
993 """
994 """
995 """
996 """
997 """
998 """
999 """
1000 """

```

GNN basé sur les Edges : Faible efficacité
 GNN basé sur les Triplets : efficacité supérieure à la combinatoire
 Entraînement : 700,000 events (~4 jours)
 Évalué sur 5000 événements
 2,7 Million Paramètres
 Inférence GPU plus complexe

Category	Metric	Allen	Etx4velo
Long, no electrons ✓ In acceptance ✓ Reconstructible in the velo ✓ Reconstructible in the SciFi ✓ Not an electron	Efficiency	99.26%	99.28%
	Clone rate	2.54%	0.96%
	Hit efficiency	96.46%	98.73%
	Hit Purity	99.78%	99.94%
Long electrons ✓ In acceptance ✓ Reconstructible in the velo ✓ Reconstructible in the SciFi ✓ Electron	Efficiency	97.11%	98.80%
	Clone rate	4.25%	7.42%
	Hit efficiency	95.24%	96.54%
	Hit purity	97.11%	98.46%
Long, from strange ✓ In acceptance ✓ Reconstructible in the velo ✓ Decays from a strange Good proxy for displaced tracks	Efficiency	97.69%	97.50%
	Clone rate	2.50%	0.92%
	Hit efficiency	97.69%	98.22%
	Hit purity	99.34%	99.68%
X	Ghost rate	2.18%	0.76%

Proceedings of the CTD 2023
 PROC-CTD2023-XX October 26, 2023
 Graph Neural Network-based pipeline for track finding in the Velo at LHCb

Résultats actuels : Comparaison Inférence C++ / Python

Exportation du NN, et du GNN sous format Torch et ONNX

Import dans Allen , plusieurs sources d'écarts possible : inférence Pytorch , simple / double precision , arrondis d'export, etc...

ETX4VELO

TrackChecker output	16/	250	6.40% ghosts			
01_velo	116/	123	94.31% (94.31%)	6 (4.92%) clones	pur 98.04%	hit eff 87.28%
02_long	72/	73	98.63% (98.63%)	4 (5.26%) clones	pur 98.08%	hit eff 87.86%
03_long_P>5GeV	51/	52	98.08% (98.08%)	2 (3.77%) clones	pur 97.25%	hit eff 89.28%
04_long_strange	3/	3	100.00% (100.00%)	0 (0.00%) clones	pur 95.24%	hit eff 100.00%
05_long_strange_P>5GeV	2/	2	100.00% (100.00%)	0 (0.00%) clones	pur 92.86%	hit eff 100.00%
08_long_electrons	5/	11	45.45% (45.45%)	0 (0.00%) clones	pur 95.00%	hit eff 81.73%

ETX4VELO_CPP

TrackChecker output	19/	250	7.60% ghosts			
01_velo	113/	123	91.87% (91.87%)	4 (3.42%) clones	pur 97.72%	hit eff 88.89%
02_long	68/	73	93.15% (93.15%)	2 (2.86%) clones	pur 98.18%	hit eff 91.21%
03_long_P>5GeV	48/	52	92.31% (92.31%)	1 (2.04%) clones	pur 97.81%	hit eff 92.15%
04_long_strange	3/	3	100.00% (100.00%)	0 (0.00%) clones	pur 95.24%	hit eff 100.00%
05_long_strange_P>5GeV	2/	2	100.00% (100.00%)	0 (0.00%) clones	pur 92.86%	hit eff 100.00%
08_long_electrons	7/	11	63.64% (63.64%)	0 (0.00%) clones	pur 92.86%	hit eff 83.89%

Merci Fotis...

Conclusions & Perspectives

Approuvé par LHCb : Grand Intérêt dans la communauté

L'efficacité pour le Velo est résolue. Reste l'inférence.

Nécessité de combiner des algorithmes **Combinatoires et Deep Learning (Le deep learning seul est un mythe)**

Le travail d'inférence pour le Velo est en cours

Plusieurs obstacles :

- Formats d'exports : Pytorch , ONNX ,
- Paralléliser les algorithmes combinatoires ; kNN et Connected Components sur GPU
- Utiliser les bibliothèques nvidia TensorRT, profilage des algorithmes
- Optimiser la tailles des réseaux pour réduire les hyperparamètres

Obstacles à venir : Générer les kernels dans Allen, lancer plusieurs kernels en parallèle selon la séquence Allen.

GNN pour le Sci-Fi en cours.

Éventuellement : un GNN global.

Merci

SUPPORT

Ext4velo Cpp (1er essais)

Etx4velo

TrackChecker output	:	15/	249	6.02% ghosts					
01_velo	:	116/	123	94.31% (94.31%),	6 (4.92%) clones, pur	97.88%, hit eff	87.03%		
02_long	:	72/	73	98.63% (98.63%),	4 (5.26%) clones, pur	98.05%, hit eff	87.55%		
03_long_P>5GeV	:	51/	52	98.08% (98.08%),	2 (3.77%) clones, pur	97.51%, hit eff	88.94%		
04_long_strange	:	3/	3	100.00% (100.00%),	0 (0.00%) clones, pur	95.24%, hit eff	100.00%		
05_long_strange_P>5GeV	:	2/	2	100.00% (100.00%),	0 (0.00%) clones, pur	92.86%, hit eff	100.00%		
06_long_fromB	:	0/	0	nan% (nan%),	0 (nan%) clones, pur	nan%, hit eff	nan%		
07_long_fromB_P>5GeV	:	0/	0	nan% (nan%),	0 (nan%) clones, pur	nan%, hit eff	nan%		
08_long_electrons	:	5/	11	45.45% (45.45%),	0 (0.00%) clones, pur	95.00%, hit eff	87.44%		
09_long_fromB_electrons	:	0/	0	nan% (nan%),	0 (nan%) clones, pur	nan%, hit eff	nan%		
10_long_fromB_electrons_P>5GeV	:	0/	0	nan% (nan%),	0 (nan%) clones, pur	nan%, hit eff	nan%		

Etx4velo_Cpp

TrackChecker output	:	81/	269	30.11% ghosts					
01_velo	:	82/	123	66.67% (66.67%),	39 (32.23%) clones, pur	79.68%, hit eff	25.66%		
02_long	:	56/	73	76.71% (76.71%),	27 (32.53%) clones, pur	80.28%, hit eff	24.85%		
03_long_P>5GeV	:	38/	52	73.08% (73.08%),	16 (29.63%) clones, pur	78.95%, hit eff	27.54%		
04_long_strange	:	3/	3	100.00% (100.00%),	1 (25.00%) clones, pur	83.33%, hit eff	36.67%		
05_long_strange_P>5GeV	:	2/	2	100.00% (100.00%),	0 (0.00%) clones, pur	91.67%, hit eff	58.33%		
08_long_electrons	:	3/	11	27.27% (27.27%),	0 (0.00%) clones, pur	66.67%, hit eff	18.10%		

LHCb / Deep Learning pour la reconstruction de traces



Premiers résultats : Dataset

Générer des datasets .csv ou parquet à partir des formats de fichier LHCb (xdigi et mdf)

Données simulées du run 3.

150Gb de données (Velo, UT, SciFi), ~500000 evenements

Average number of hits / event : 2000

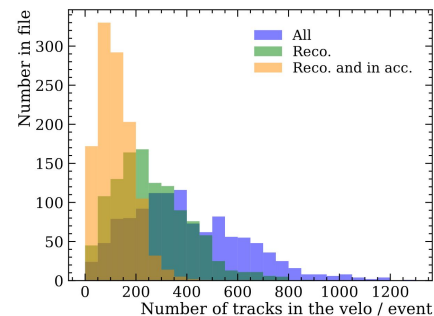
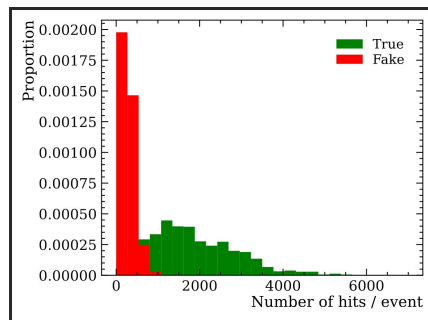
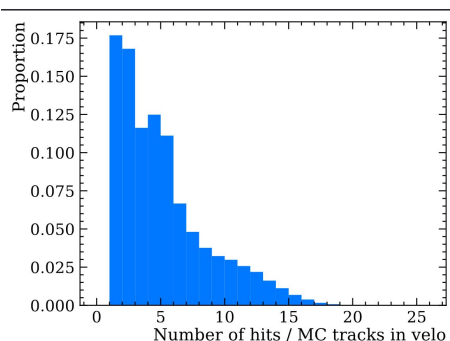
Number of MC : 400

Number of reconstructible tracks : 150

Number of needed constraints : $150 * 2 = 300$

Usually : the constraint data for learning on a best optimized neural net $1000 * \text{Number of constraints} = 300000$ Hits (This has to be evaluated for GNN)

We need $300000/150 = 2000$ events to make the learning

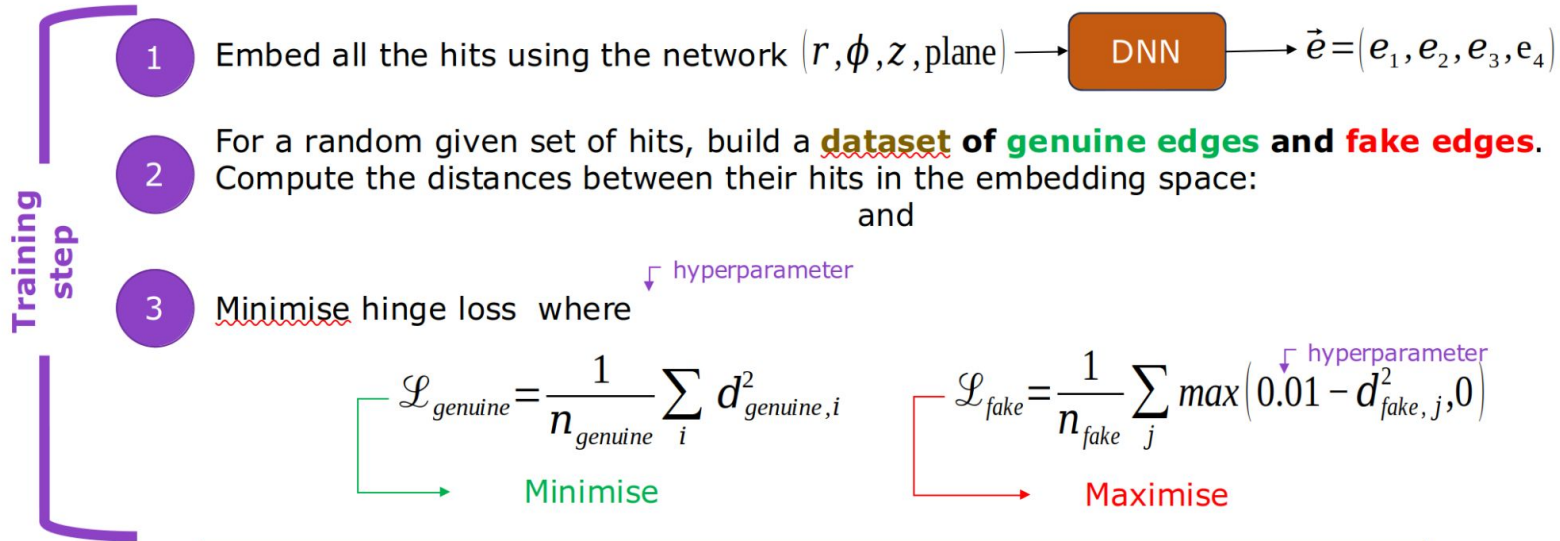


LHCb / Deep Learning pour la reconstruction de traces

Graph Building

GNN: filter edges

Build tracks from graph



Training dataset

- **Hard Negative Mining:** edges built by a kNN ("hard" negatives)
- **True** edges
- **Random** edges

Approche du problème de reconstruction

Difficultés : Caractéristiques du détecteur LHCb

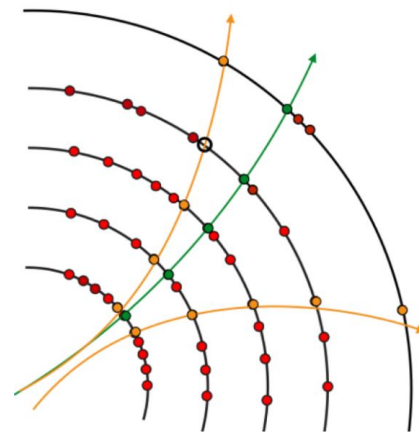
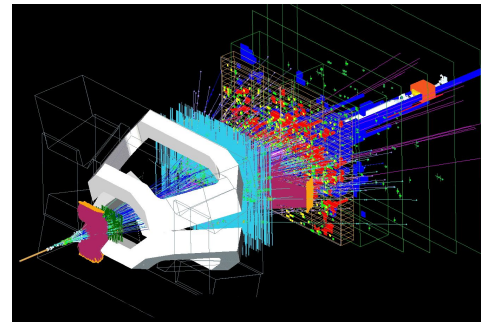
Détecteur Hétérogène : Velo (Pixel), UT , SCifi (strip) => Plusieurs échelles de granulométrie du détecteur. Efficacité de reconstruction plus faible dans la zone de collision.

Forte hétérogénéité spatiale du phénomène de collision :

=> discontinuité spatiale (NN actuels sur données 2D ou 1D mais continues)

Collision = Séquence => nécessité d'utiliser une approche par récurrence (actuellement : filtres de Kallmann)

Problème 3D = grande dimensionnalité , pas très courant en deep learning.



Prospectives pour le Deep learning

90 % des activités dans le deep learning : Traitement de l'image / traitement du langage / marketing et modèles de suggestion. Domaine en rapide évolution.

MLops

=> Problèmes liés à l'ingénierie logicielle : reproductibilité et exigence scientifique ,
versionning du Code / Datasets , DagsHub

Techniques propres au deep learning

Librairies (Tensorflow / Keras , [Pytorch](#)) : développement d'outils propres à partir de ces librairies (exp : convolutions particulières)

[GNN](#) (graph Neural networks) , tendance forte du domaine, Combiné au Spectral (LHCb)

Génération de données avec les GANs (alternatives à Geant4) moins gourmands en ressources, ou [Diffusion Models](#)

Combinaison méthodes spectrales / deep learning

[PINNs](#) (physics Informed Neural Networks) => en interaction forte avec les physiciens

Hardware

Architecture hybride CPU/GPU mémoire séparée (actuels)

GPU embarqués ([nvidia Nano](#)) (cf groupe de travail Think)

Architecture arm / GPU à mémoire partagée (apple M1/M2)

[FPGAs](#) dédiés au deep learning / Interaction avec les électroniciens (cf groupe de travail Think), en cours avec le [LIP6](#)



Critère: l'énergie
Technologie future
incertaine



Activités en cours sur le Deep Learning

ATLAS : Jet classification by Neural Nets , the Ringer project.

unfolding , Photons isolation and Pile-ups , GANs, uncertainties,

GRAND : Smart Trigger avec réseaux de neurones

COMET :Electron drift with CNNs



Activités séparées, des physiques différentes,
mais les outils techniques sont communs