Hyperparameter tuning for FastCaloGan

Alkaid Cheng, Michele Faucci Giannelli, Matthew Barber, Mikel Zemborain, Rui Zhang

(With support from the PanDA/iDDS team & FCG team)

University of Wisconsin-Madison, Wisconsin

Machine Learning for Simulation meeting

Oct 28, 2021

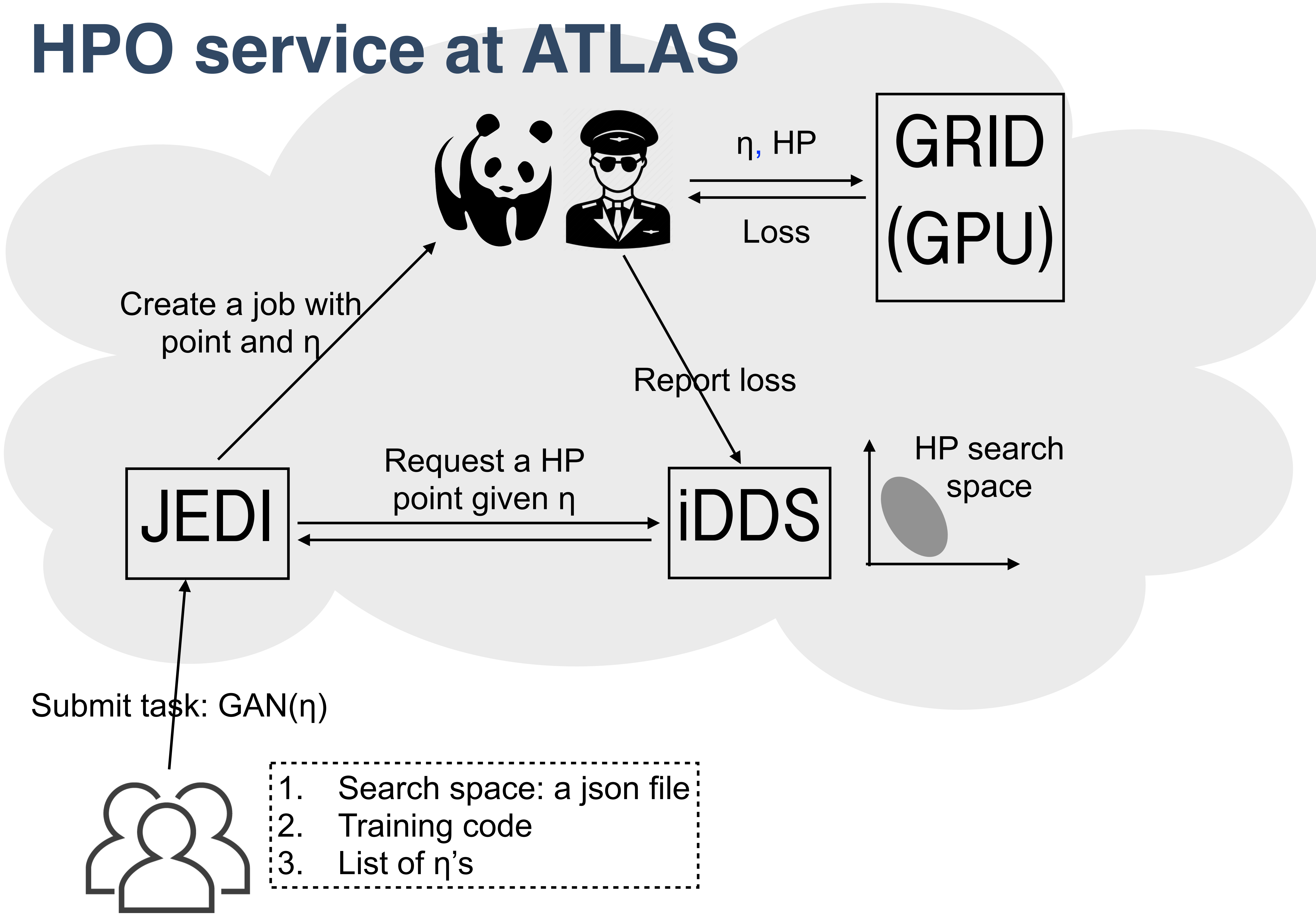
Introduction to FastCaloGan

- GAN — A ML solution for showering simulation in calorimeter
  - Previous presentation @ ML4Sim by Michele
  - 100 η slices × 3 particles — 300 GANs to train
  - Each GAN train ~1M iterations and select the best iteration
  - Challenging hyperparameter optimisation due to large search space and long training time
- Now used in AF3 for pion simulation in mid energy range [16–256] GeV
  - Pub note, AltFast3 paper
  - No HPO was done for v1 (what is in AF3)
  - Time allowed for fine tuning the model before commissioning
- Today:
  - Hyperparameter optimisation (HPO) + improvements on ML side

New training strategy

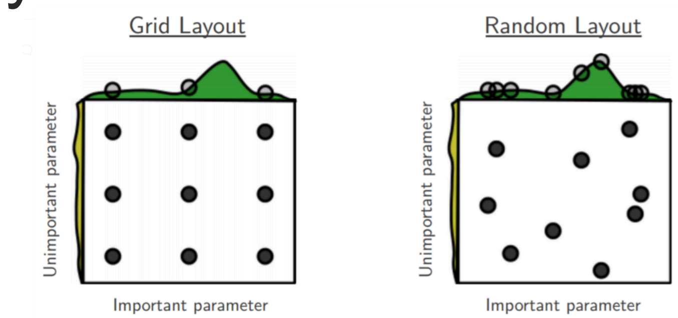
- The total training time of GANs is significant
  - ×N for HPO
- It is possible to exploit transfer learning
  - A “seed” GAN is trained for long time in each region, then other GANs in the same region start from the best iteration of the seed GAN and hopefully need less iterations.
  - The strategy worked
- It is possible to speed up training
  - In a photon GAN, the GPU utilisation was ~30% → not fully using the GPU power
  - Code was rewrote in native TF2 syntax, using TF data loader
  - Try to prepare all data at once on CPU, to prevent CPU-GPU communication during training
  - New code speed up the photon GAN by 2×, pion GAN by 5× (pion GAN is larger therefore the delay from earlier was larger)

HPO service at ATLAS



HPO with a large space

- Given the long training time, **random** search is used
  - More time efficient than other sequential search such as Bayesian
  - More coverage than Grid search



- Search space definition
  - Five HPs are considered: two lr's for G and D, two β's in both Adam, batch size

HP	Method	Range
G_lr, D_lr	loguniform	(10-5, 10-3)
G_β, D_β	loguniform	(0.55, 0.99)
Batch size	categorical	[128, 256, 512, 1024, 2048]

Photons: HPO

- G_lr: 10^{-5} — $10^{-4.5}$
  - Then, not much constraints in G_β
- D_lr: 10^{-5} — $10^{-4.5}$
  - Then, not much constraints in D_β
- Momenta in Adam optimiser doesn’t have big impacts
  - Probably because there is a pre-training step
- Batch size: 1000–1500 looks like a general good choice

Photons: Initialisation algorithm

- "glorot_normal" was used for neuron weights initialisation
- Expected because He initialiser works better for layers with ReLu activation. And we use ReLu.

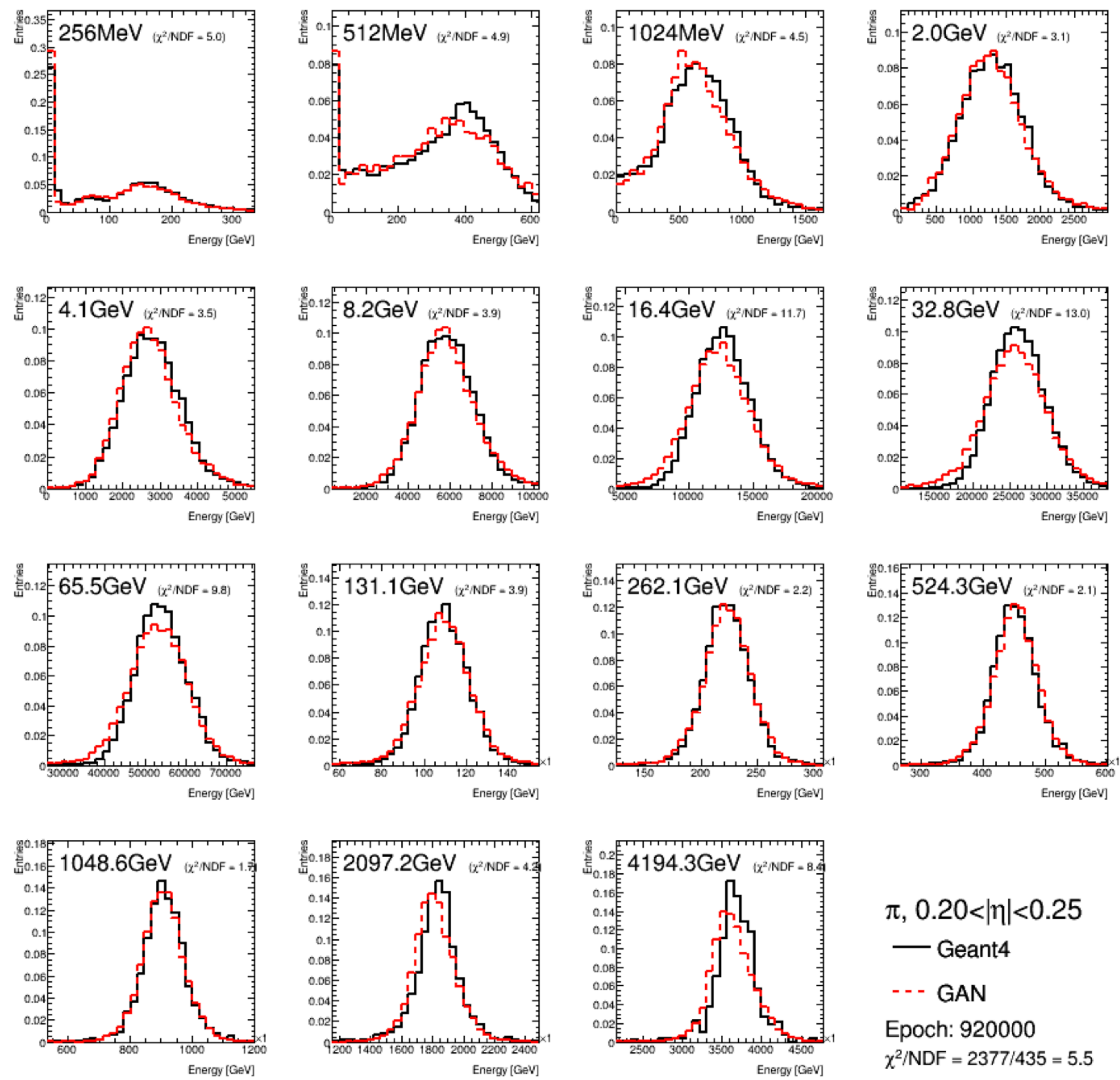
Initialisation	0.00–0.05	0.20–0.25	0.60–0.65	0.80–0.85	Comments
He_normal (1M)	4—6	5—9			A few failure for 0–0.05
glorot_normal (1M)	2—4	5—10	χ^2		A few failure for 0.20–0.25
he_uniform (1M)	3—6	5—8			Good for both η
glorot_uniform (1M)	2—4	11–20			Failed for all 20-25
he_uniform (2M)	2—3	4—5	6—10	4—5	2 runs only

Iterations

- **he_uniform** seems to work best

Pions: Optimisation of generator

100-200-400-800*

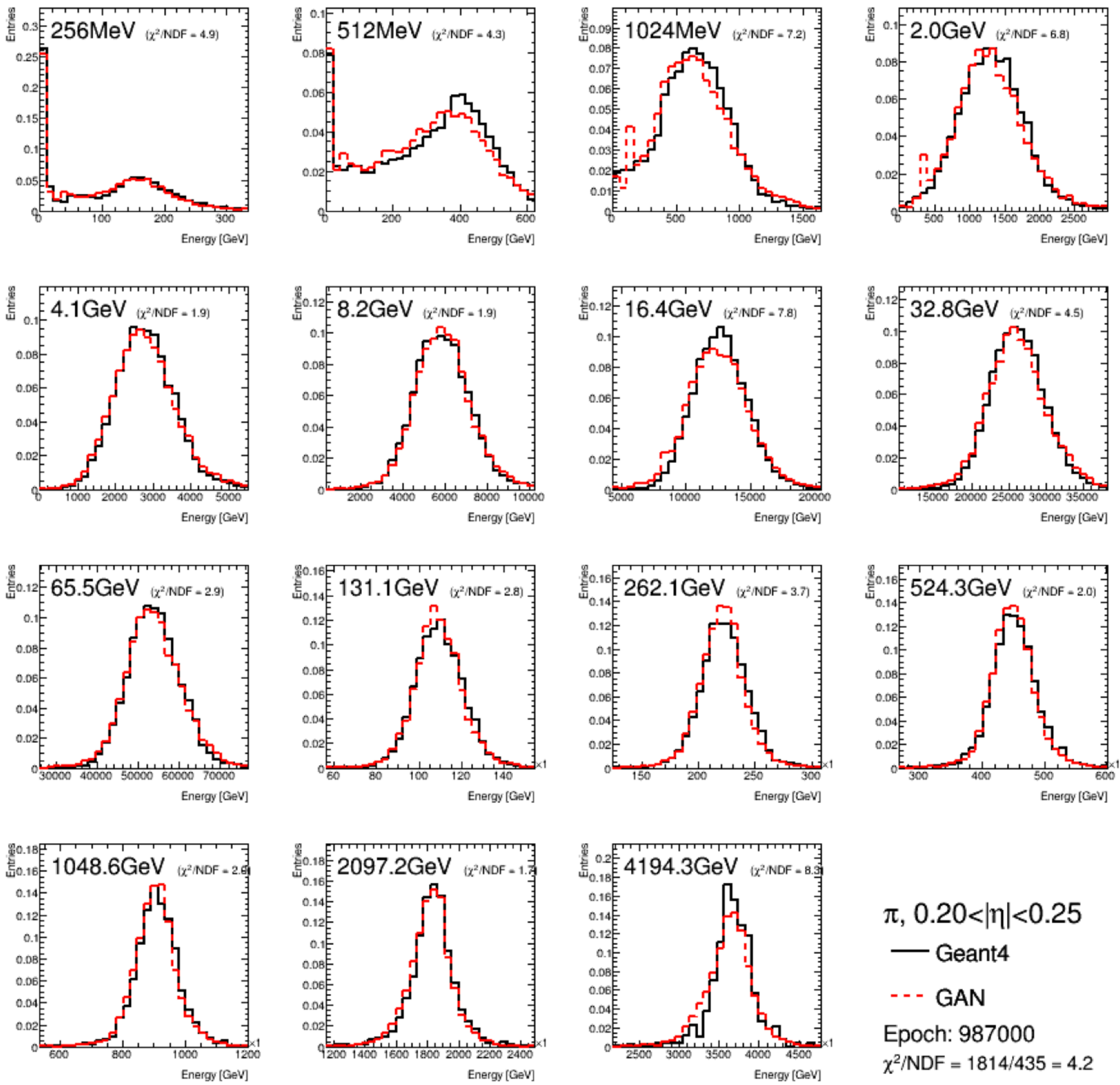


	50-50-100-100	100-100-150-200	100-128-256-512	**100-200-400-800**	100-256-512-1024
χ^2	25.3	24.2	8.2	**5.5**	5.9

* “Latent dimension”-“first layer size”-“second layer size”-“third layer size”

Pions: Optimisation of discriminator

800-400-200*



	1086-1086-1086	800-400-200	**800-400-200**
χ^2	5.5	6.6	**4.2**

* "Latent dimension"-"first layer size"-"second layer size"-"third layer size"

Pions: Others optimisation

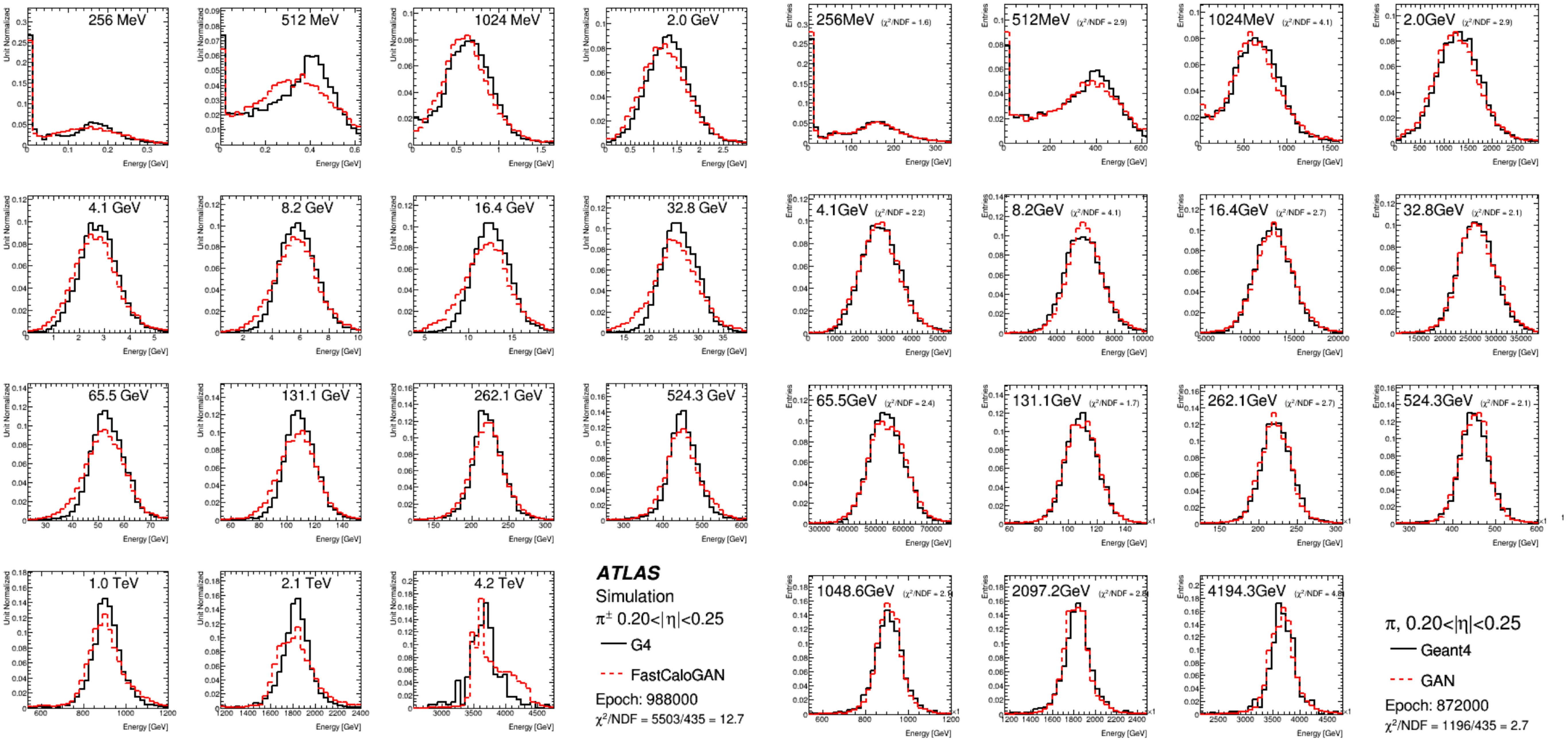
- Energy normalisation

$$\hat{E} = \frac{E}{E_{\max}} \Rightarrow \hat{E} = \frac{\log\left(\frac{E}{E_{\min}}\right)}{\log\left(\frac{E_{\max}}{E_{\min}}\right)}$$

- Changed latent space from Uniform to Normal
- Increase batch size from 128 to 512
  - χ^2: 4.2 to **2.7** (Further increasing BS doesn't gain much)
- Discriminator/Generator Training Ratio
  - 5 → 3: no performance gain/loss but reduce training time
- Further details about pion optimisation can be found in this report.

Pions: Summary

AF3 Paper | Now



χ^2 improved from 12.7 to 2.6, similar training time, double voxels

Conclusions

- FastCaloGAN is improved significantly from the AF3 paper results
  - Since then a lot of improvements, eg HPO, Voxelisation binning doubled (536 to 1086)
  - Large speed up → can train with larger iterations → better performance
  - π χ^2: 12.7 to 2.6; γ χ^2: 13.5 to 3.8
  - Lower χ^2 indicates better performance as shown in the plots, more validations performed but not included today.
- Various places to improve for the future
  - Low energy pions
  - Large eta photons
- **It is promising and very possible to see more and more GANs reach <10 χ^2 in the near future**