Belle II ML effort at TAU

Ori Fogel, Ran Gilad-Bachrach, Abi Soffer

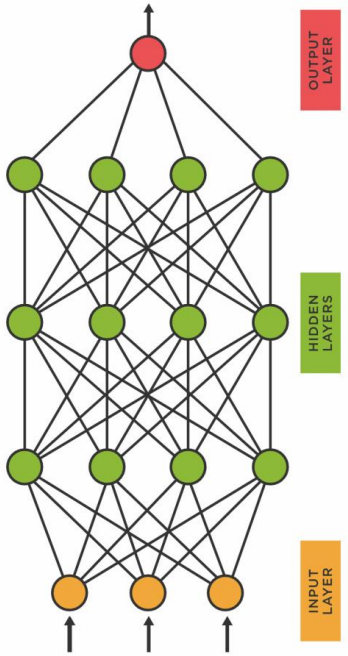
Tel Aviv University

Outline

- Fixed-input NNs vs. the DeepSets architecture
- MultiDeepSets (MDS) architecture
- Continuum-background suppression
- Foundation model (FM) for Belle II

Fixed-input neural nets

- A MLP is a function $f(x_1, \ldots, x_{N_V})$ that takes N_V inputs
- In a particle-physics event, the number of particles N_P can vary event-by-event
- So physicists "pool" information from the N_P particles to calculate N_V variables that we deem useful
- E.g., missing energy and momentum in "CLEO" cones:



$$E_{\text{miss}} = \sqrt{s} - \sum_{p=1}^{N_p} E_p\,, \qquad C_b = \sum_{\theta_p \in [\theta_b, \theta_{b+1}]} p_p$$

- Disadvantage: some information is lost
- Mitigation: use enough input variables to hopefully recover lost information
- But it's better to use relevant variables of individual particles

DeepSets

- Use of the DeepSets architecture utilizes the fact that particles are an unordered (permutation-invariant) set
- It's a function of the form

$$f(\{x_p\}) = \rho(P(\{\phi(x_p)\}))$$

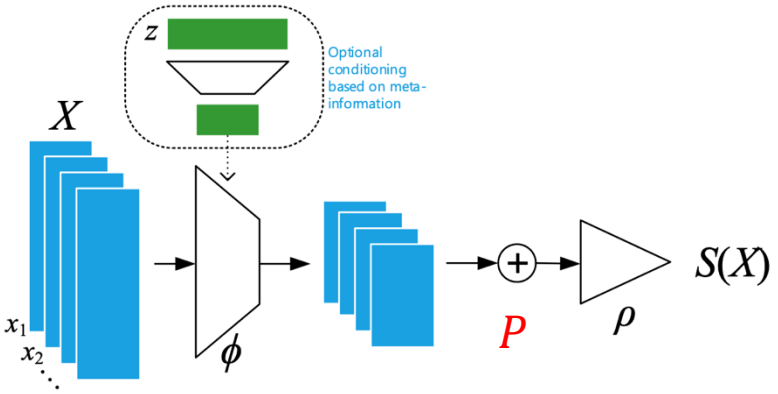
MLP

P pools information from all the particles, e.g., mean:

MLP

x_p = array of variables for particle p

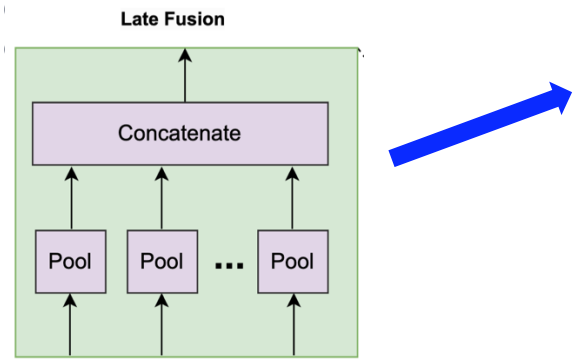
$$P(\{\phi(x_p)\}) = \frac{1}{N}\sum_{p}^{N} \phi(x_p)$$

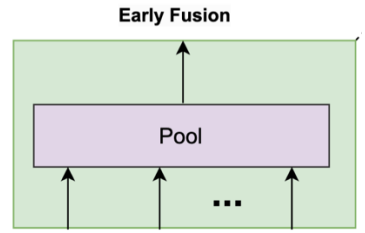


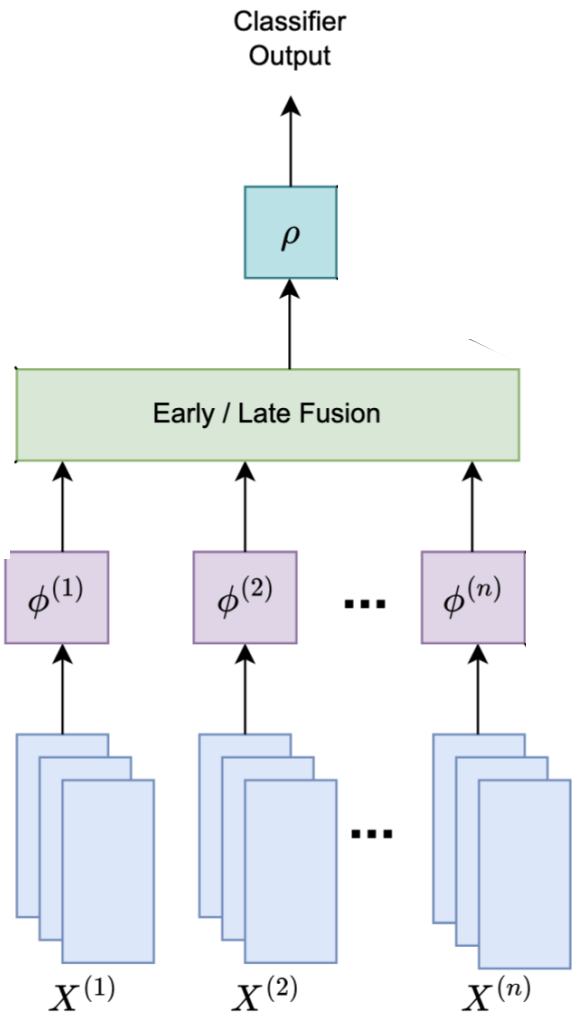
- Advantage: pooling is done in the "latent space" (output of $\phi(x_p)$), allowing determination & retention of useful information
- Much use in particle physics

MultiDeepSets (MDS)

- Many applications involve more than one set.
- E.g., tracks & photons have different variables (and unequal numbers of variables)
- → We created MDS
- We use several DS blocks and fuse them before or after pooling



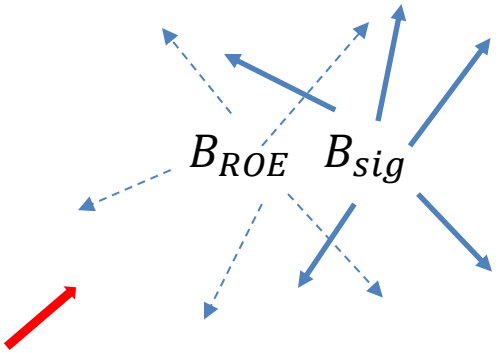




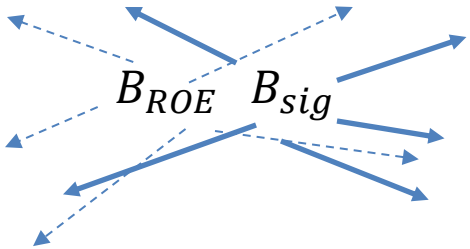
Surprisingly, this straightforward solution isn't discussed in the ML literature

Continuum suppression at Belle II

- In a B-physics analysis at a $e^+e^- \to B\bar{B}$ experiment, one typically reconstructs the signal B_{sig} and ignores the rest of the event (ROE)



- A common task is suppression of $e^+e^- \to q\bar{q}$ ("continuum") background, where $q = u, d, s, c$
- Exploited features:
  - B = heavy, slow, have no spin → roughly isotropic decays
  - q = light fermions → roughly 2-jet distribution with $1 + \cos^2\theta$ distribution



- MVAs first exploited in the 1990s (AFAIK)
- Currently at Belle II: combine 30 variables using a BDT, including
  - CLEO cones
  - Momentum-weighted Legendre coefficients of 2-particle angular separations

MDS for continuum suppression

- We use 5 sets of particles, with the following variables for each particle:

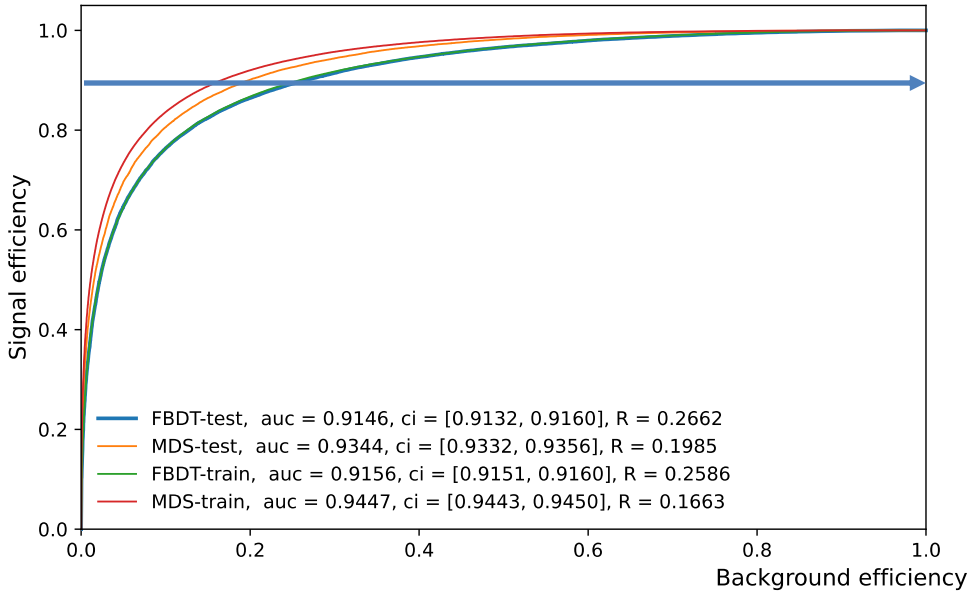
SET	MAX PARTICLE NUM	FEATURES
ROE TRACKS	10	$\vec{p}, q, I^{(h)}, dr, dz$
ROE PHOTONS	20	$\vec{p}$
B^{SIG} TRACKS	10	$\hat{p}, q, I^{(h)}, dr, dz, p_{scale}$
B^{SIG} PHOTONS	20	$\hat{p}, p_{scale}$
B^{SIG}	1	$\hat{p}, \vec{T}$

Thrust vector

For B_{sig} children, replace p with

$$p_{\text{scale}} = \frac{p}{\max p}$$

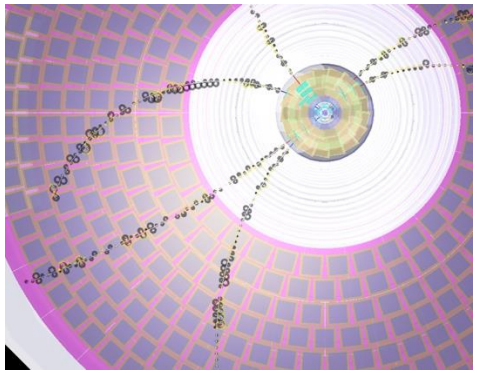
to avoid correlation with $M_{bc} = \sqrt{\frac{s}{4} - p_B^2}$

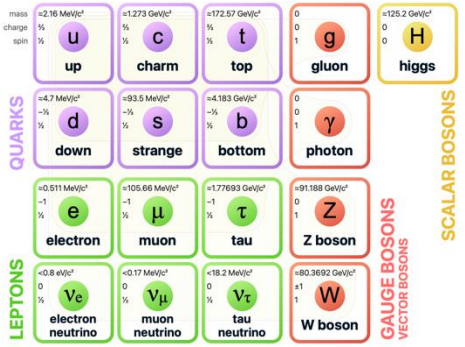


- For a benchmark signal efficiency of 90%, we get 25% less background
- Performance could degrade on data due to data-MC diff – yet to check.
- There are methods for dealing with that, hopefully won't be needed

Training and generalization

- In the usual continuum-suppression application there is only one (or a few related) signal B decay mode
- In our training, we take signal from the “full event interpretation” module, which attempts to reconstruct thousands of modes.
- We have yet to check what happens on specific modes
- But perhaps this multi-mode training teaches our algorithm to work with any mode, including modes it hasn’t been trained on
- If this is the case, users could use it for their modes without retraining





Foundation model (FM) for Belle II

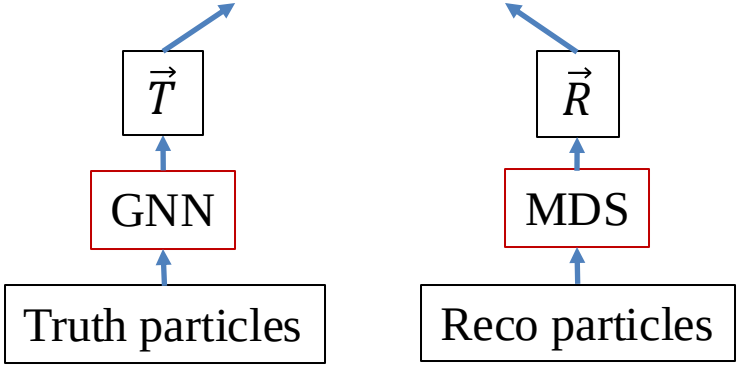
- A FM is a ML algorithm ("model") that trained on enough general information about a system that it can perform a specific "downstream task" with very little (or even without) training for this task.
  - E.g., LLMs (ChatGPT, etc.)
- Several recent works in particle physics [1, 2, 3, 4]
- In particle physics, the solution to all tasks is the truth-level process
  - Therefore, a model that trained on enough examples may be able to perform a specific new task, e.g., signal/background classification in a new decay mode
- We are just starting with this project

Potential method 1: supervised contrastive learning

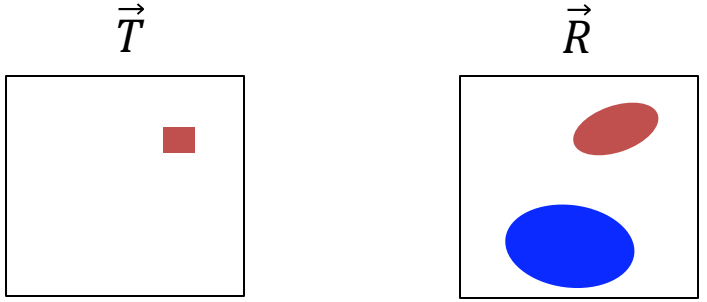
$$\min \frac{\vec{T}_i \cdot \vec{R}_j}{\vec{T}_i \cdot \vec{R}_i}$$

Latent representation:

Original representation:

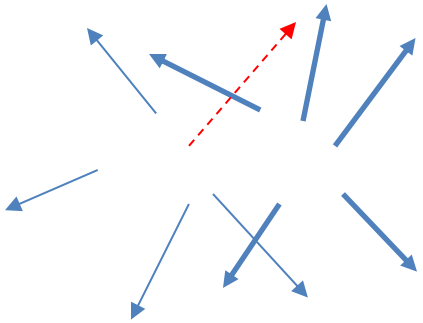


After training, correct reco-truth association leads to similar placement in the latent space:



Potential method 2: self-supervised learning with masking

Hide some particle(s),
train model to predict the missing particle(s).



Inspired by LLM training methods.

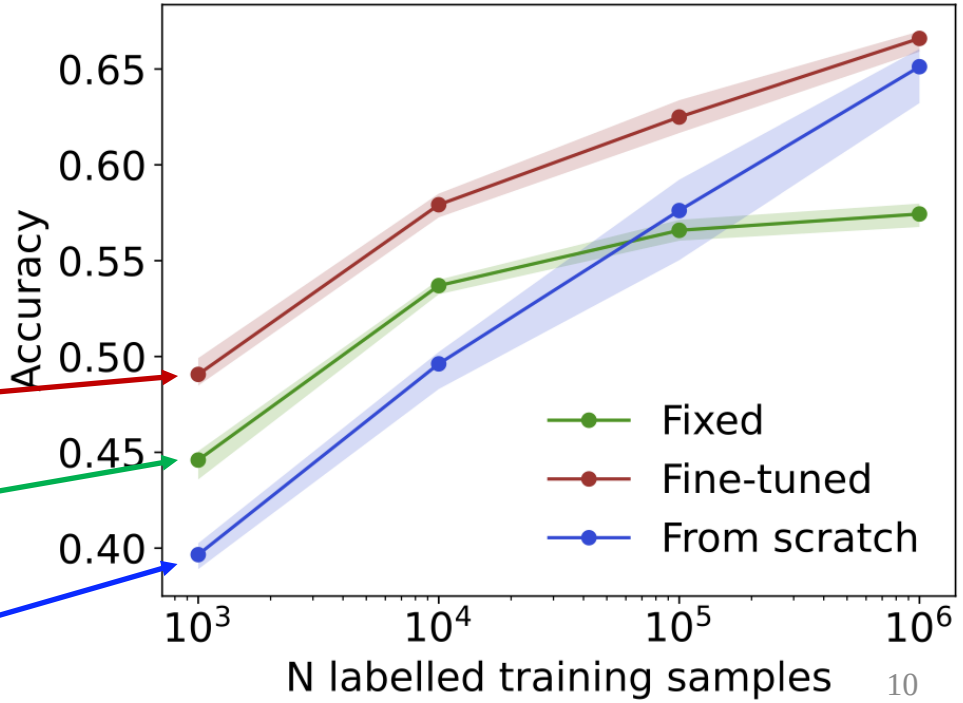
Can be trained with detector data
(but performance might be inferior to MC training)

This approach has been studied in Ref. 4 for jet classification, showing improved performance:

Retrain the entire NN starting from the pretraining parameters

Train only part ("head") of the NN

From-scratch training for downstream task



Thank you!

Backup slides

Variables used in standard continuu, suppression

- 'R2',
- 'thrustBm',
- 'thrustOm',
- 'cosTBTO',
- 'cosTBz',
- 'KSFWVariables__boet__bc',
- 'KSFWVariables__bomm2__bc',
- 'KSFWVariables__bohso00__bc',
- 'KSFWVariables__bohso02__bc',
- 'KSFWVariables__bohso04__bc',
- 'KSFWVariables__bohso10__bc',
- 'KSFWVariables__bohso12__bc',
- 'KSFWVariables__bohso14__bc',
- 'KSFWVariables__bohso20__bc',
- 'KSFWVariables__bohso22__bc',
- 'KSFWVariables__bohso24__bc',
- 'KSFWVariables__bohoo0__bc',
- 'KSFWVariables__bohoo1__bc',
- 'KSFWVariables__bohoo2__bc',
- 'KSFWVariables__bohoo3__bc',
- 'KSFWVariables__bohoo4__bc',
- 'CleoConeCS__bo1__bc',
- 'CleoConeCS__bo2__bc',
- 'CleoConeCS__bo3__bc',
- 'CleoConeCS__bo4__bc',
- 'CleoConeCS__bo5__bc',
- 'CleoConeCS__bo6__bc',
- 'CleoConeCS__bo7__bc',
- 'CleoConeCS__bo8__bc',
- 'CleoConeCS__bo9__bc'