

Event Classification with Multi-step Machine Learning

18 / 5 / 2021
vCHEP'21 @ online

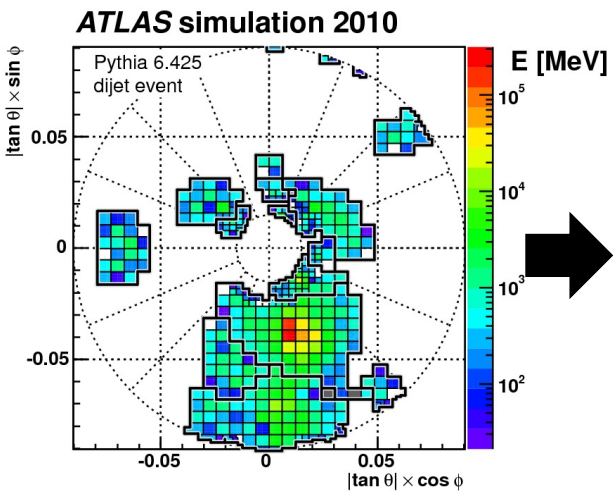
M. Saito^{1,3}, T. Kishimoto^{1,3}, Y. Kaneta², T. Itoh², Y. Umeda²,
J. Tanaka^{1,3}, Y. Iiyama¹, R. Sawada¹, K. Terashi¹

International Center for Elementary Particle Physics¹, BrainPad Inc.², Institute for AI and Beyond³

Motivation

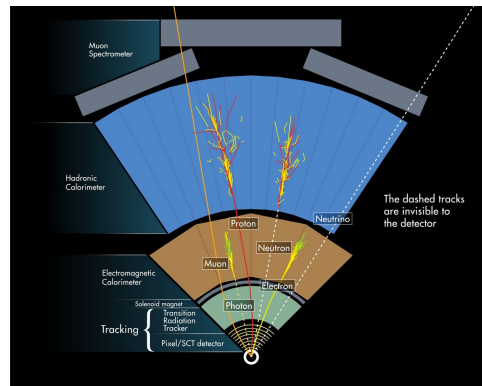
- The application of deep learning in the HEP is mature
 - for specific tasks and the **single task**
- Most of our tasks consist of small sub-tasks

CaloClustering/Tracking



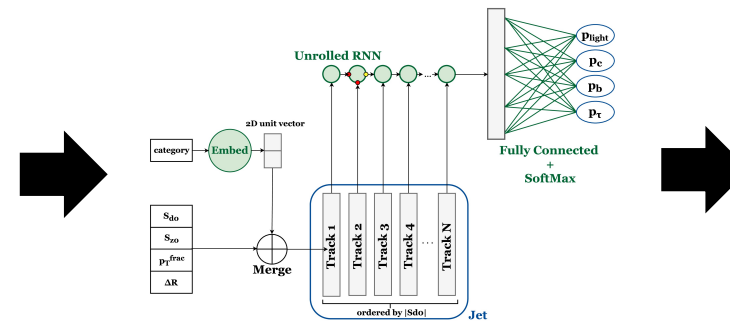
[Eur. Phys. J. C 77 \(2017\) 490](#)

Particle reconstruction



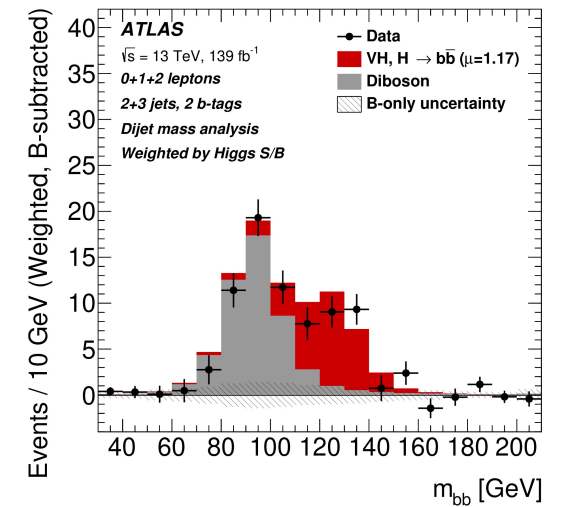
[CERN-EX-1301009](#)

Particle identification



[ATL-PHYS-PUB-2017-003](#)

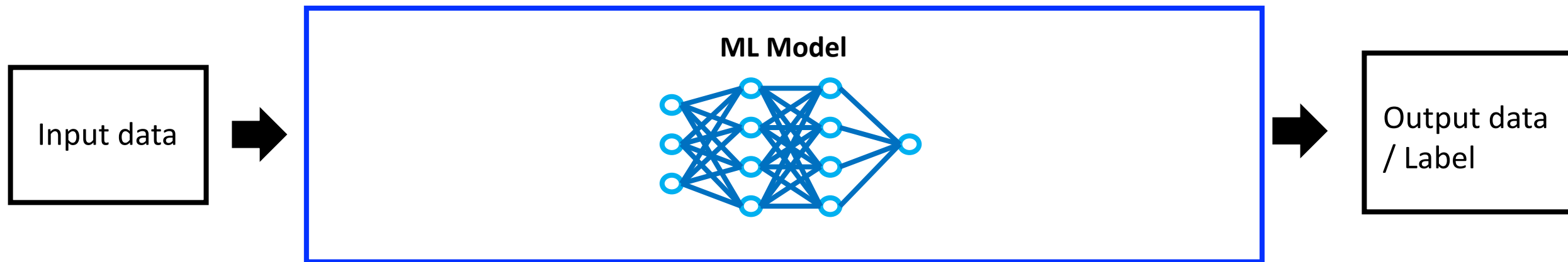
Physics analysis



[Eur. Phys. J. C 81 \(2021\) 178](#)

Motivation

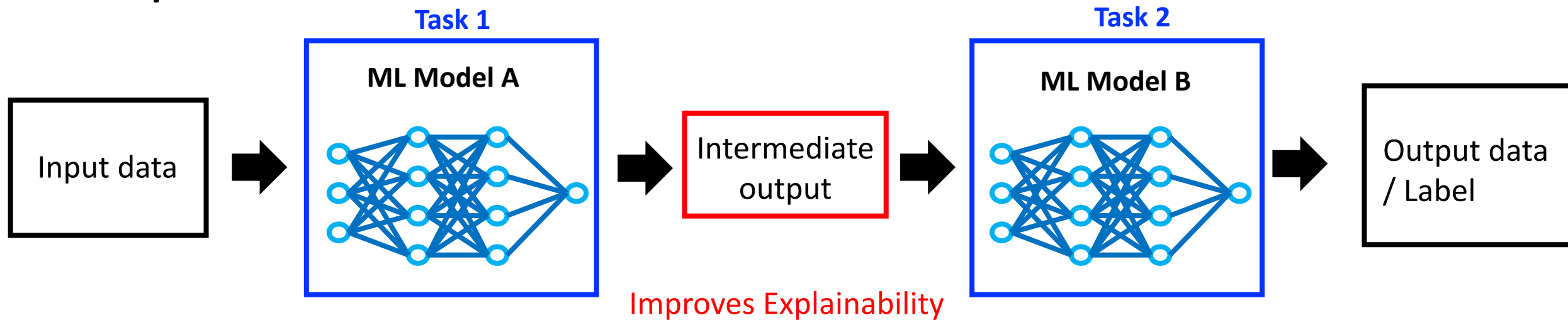
- The application of deep learning in the HEP is mature
 - for specific tasks and the **single task**
- Most of our tasks consist of small sub-tasks
- Deep learning can learn such tasks end-to-end (by large single task)
 - Need huge training data / compute resources if the model is complex
 - Blackbox



Motivation

- We propose ***Multi-step ML***
 - Users define sub-tasks / intermediate output based on domain knowledge
 - physicists know how data were generated
 - Our framework (***Multi-step ML***) aims to orchestrate sub-tasks
 - find the best model parameters (model weights)
 - find the best hyperparameters

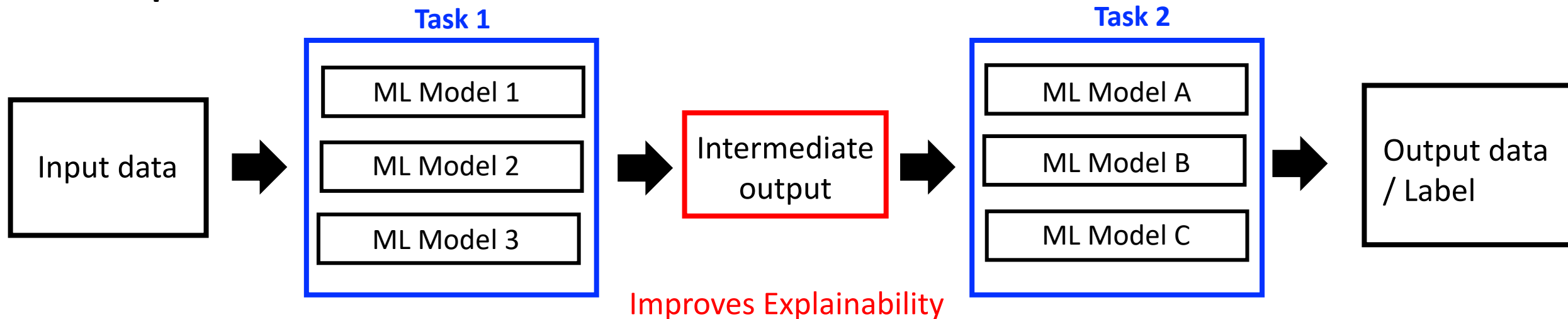
Multi-step AI



Motivation

- We propose **Multi-step ML with Model Selection (MS)** in this talk
 - Users define **several** sub-tasks / intermediate output based on domain knowledge
 - physicists know how data were generated
 - Our framework (**Multi-step ML**) aims to orchestrate sub-tasks
 - find the best model parameters (model weights)
 - find the best hyperparameters
 - find the best combination of each model

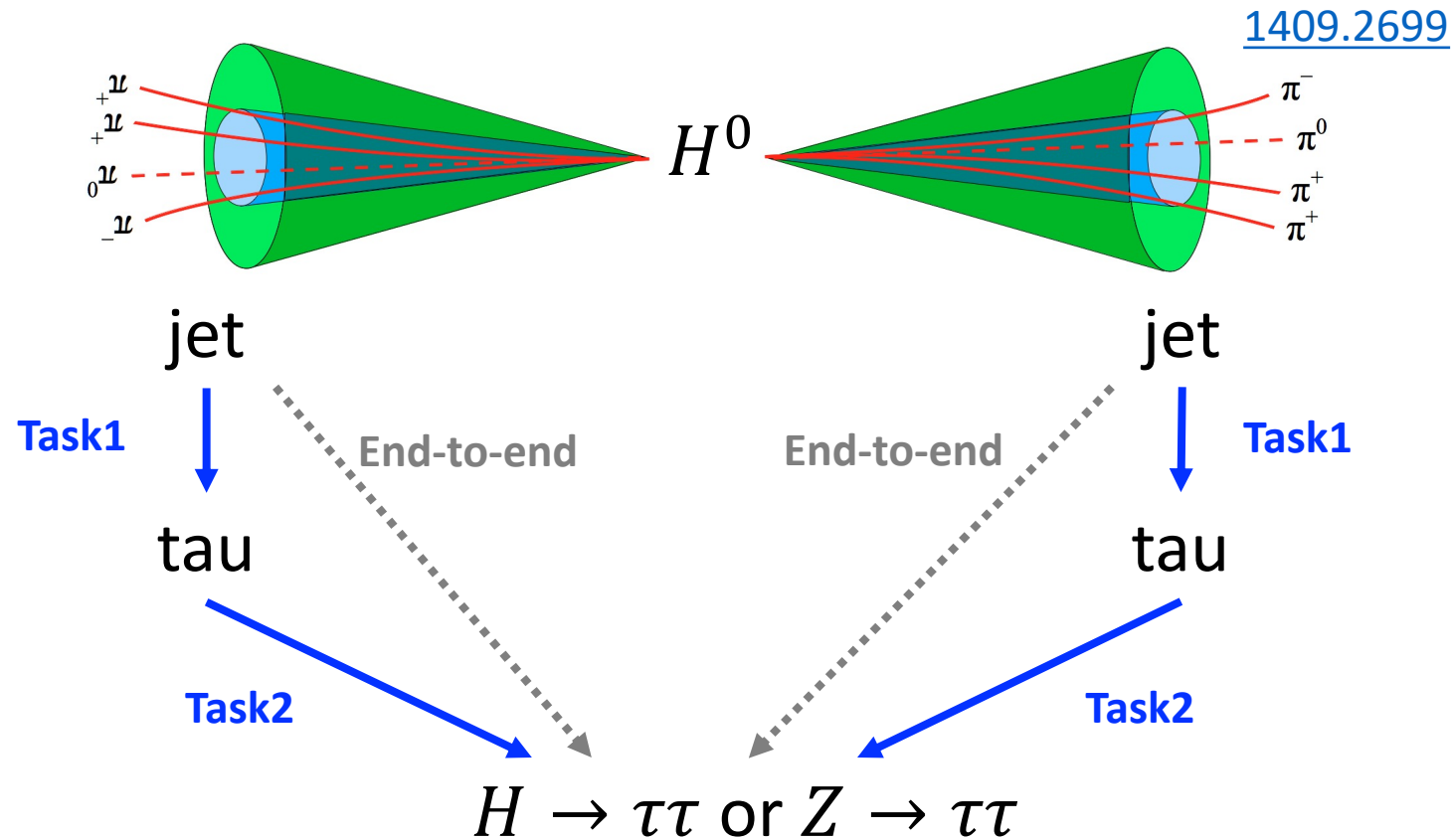
Multi-step AI with MS



Experiments: Problem

Samples are generated using Pythia8 + Delphes (100 K events)

- Classification of $H \rightarrow \tau\tau$ (signal) and $Z \rightarrow \tau\tau$ (background)
 - Task1 : Calibration of jet 4-vector (reco-level \rightarrow truth-level)
 - Task2 : Event classification (two tau 4vec \rightarrow Higgs or Z)

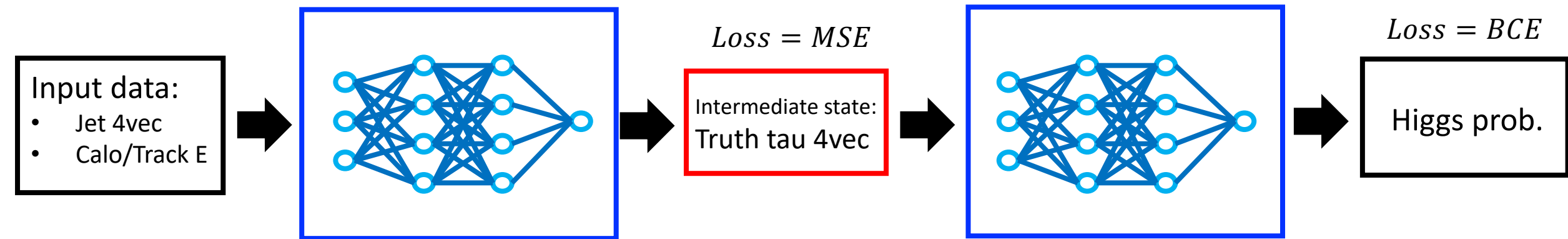


Experiments: Problem

Samples are generated using Pythia8 + Delphes (100 K events)

- Classification of $H \rightarrow \tau\tau$ (signal) and $Z \rightarrow \tau\tau$ (background)
 - Task1 : Calibration of jet 4-vector (reco-level \rightarrow truth-level)
 - Task2 : Event classification (two tau 4vec \rightarrow Higgs or Z)

Multi-step ML



Loss function is defined as a linear sum of each task's loss

$$\mathcal{L} = v_{\text{Task1}} \mathcal{L}_{\text{Task1}} + v_{\text{Task2}} \mathcal{L}_{\text{Task2}}$$

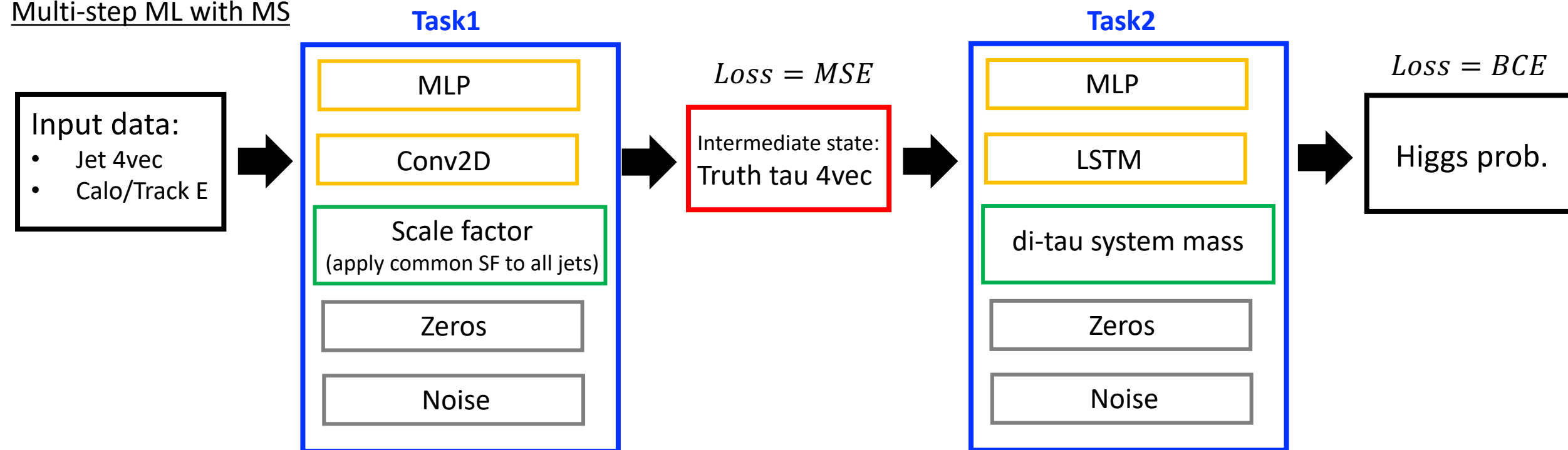
$$v_{\text{Task1}} + v_{\text{Task2}} = 1$$

Experiments: Models

We prepared five models for each task

- two DL models (MLP, Conv2D, LSTM)
- one traditional/simple model (SF, Mass)
- two dummy models (Zeros: $f(x) = 0$, Noise: $f(x) = N(0, I)$)

Multi-step ML with MS



Models are connected differentiable. We optimize all models simultaneously.

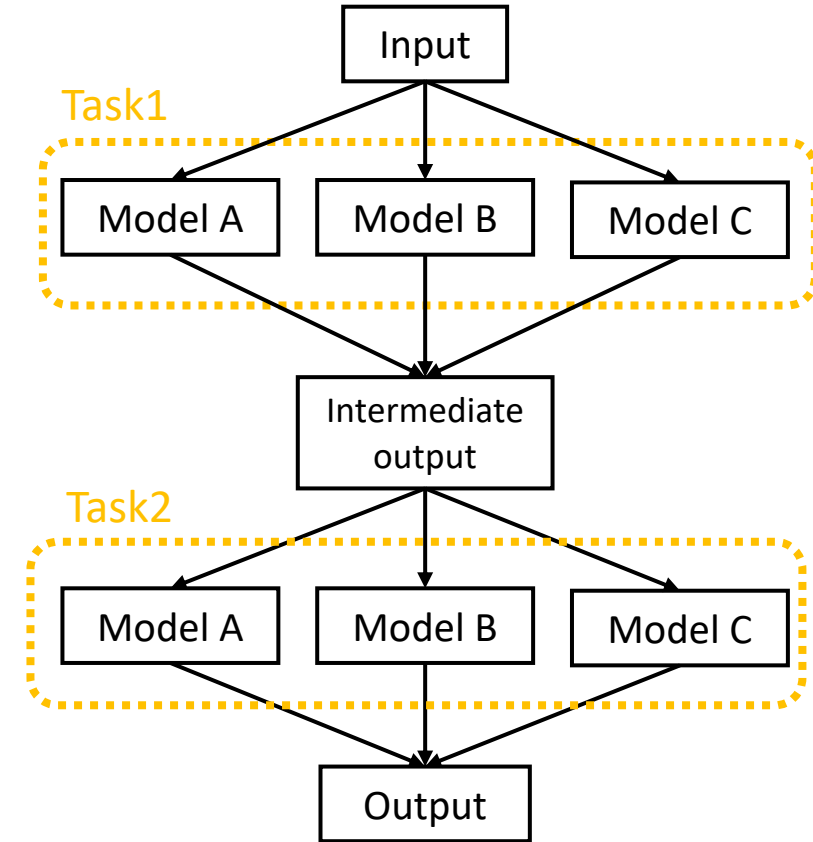
Method for model selection

- What to do for model selection
 1. Update model parameter (w_i)
 2. Select the best model combination

- Brute-force method (Grid search)

Algorithm

- For all model combination:
 - Update model parameter (w_i)
 - Evaluate performance for validation data
- Select the best model combination

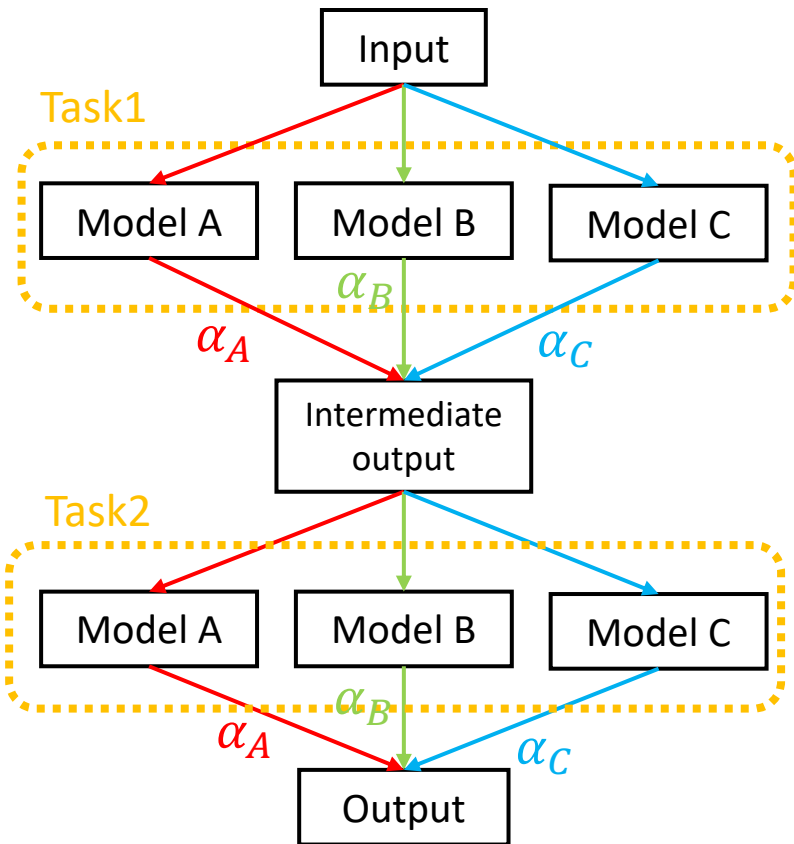


- Computational complexity :
 - $O(\prod_t N_{\text{models},t})$ (general case)
 - $O(N_{\text{models}}^2)$ (our toy model (two tasks))

- We propose other methods inspired by Neural Architecture Search (NAS)
 - [Differential Architecture Search \(DARTS\)](#) (arXiv:1806.09055)
 - [Single Path One-Shot NAS \(SPOS-NAS\)](#) (arXiv:1904.00420)

Method for model selection

Differentiable Architecture Search (DARTS)



- Assign weight (α_i) for each model
- Output is weighted sum of them

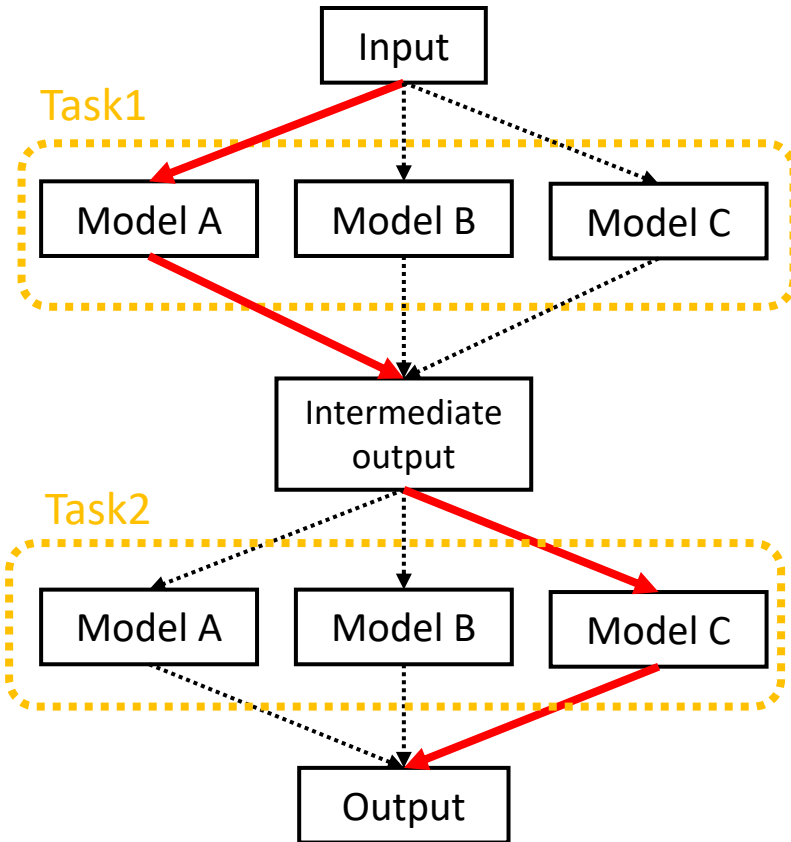
$$y_t = \sum_{i \in \text{Models}} \text{softmax}(\alpha_i) \cdot y_{t,i}$$

Algorithm

- Until convergence:
 - Update α_i using validation data
 - Update model parameter (w_i) using training data
- Select the largest α_i for each task

Method for model selection

Single Path One-Shot NAS (SPOS-NAS)



- A randomly selected model is used as the output

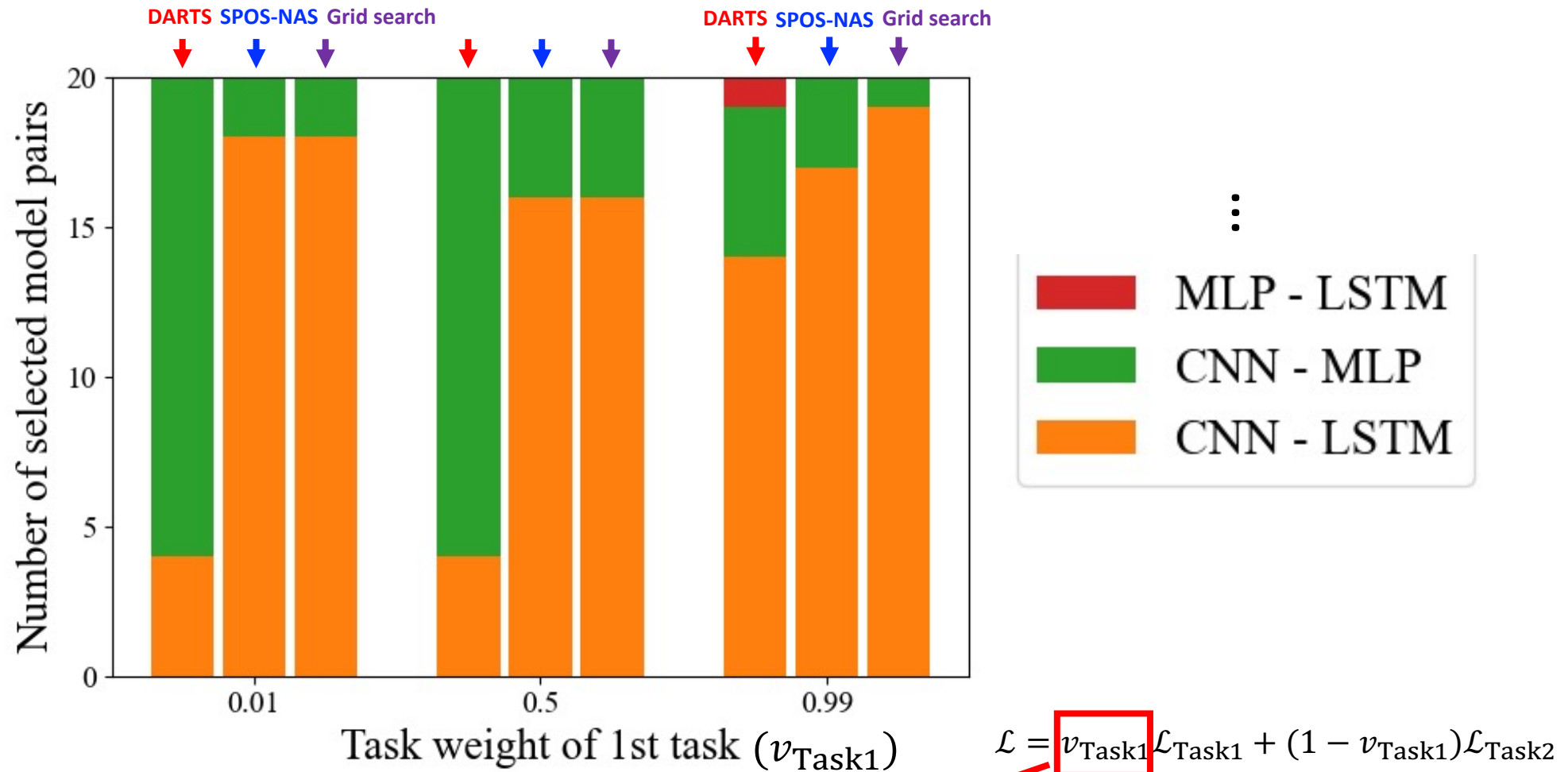
$$y_t = y_{t,i}, (i \sim \text{Uniform}[A, B, C])$$

Algorithm

- Until convergence:
 - Sampling one model in each batch
 - Update model parameter (w_i)
- Select the best model combination by evolutionary algorithm
 - For simplicity, we use grid search

Result: Model selection

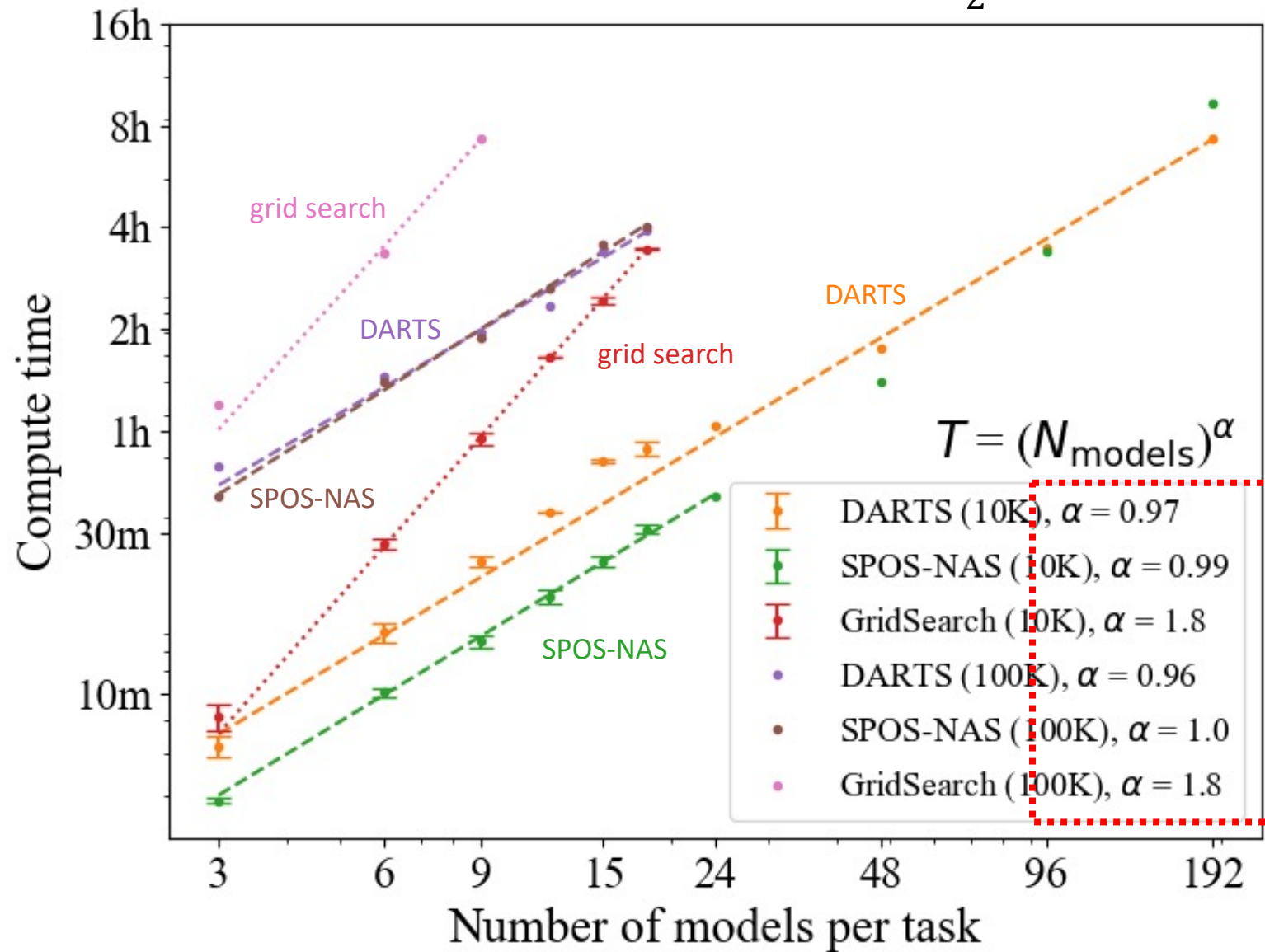
Selected model pairs



- SPOS-NAS has a similar model selection power and prediction performance as grid search
- DARTS selects similar model pairs but with a different fraction, resulting in worse prediction

Result: Scalability

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{\text{Task1}} + \mathcal{L}_{\text{Task2}})$$



NAS-based: $O(N_{\text{models}})$

Grid search: $O(N_{\text{models}}^{\text{steps}}) = O(N_{\text{models}}^2)$

Summary

- We proposed multi-step ML
 - consists of several differentially connected sub-tasks.
 - simultaneously train overall ML models.
- Model selection is performed by using the idea based on NAS (DARTS and SPOS-NAS).
 - selected models are consistent with grid search
 - our method is scalable
- Future work:
 - Extend to hyperparameter optimization
 - improve explainability
 - Apply to other problems, including other than HEP
- [Code](#) is publicly available