加速器実験における転移学習の応用

大学共同利用機関法人
高エネルギー加速器研究機構

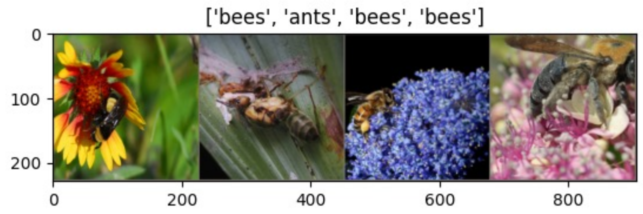
Tomoe Kishimoto

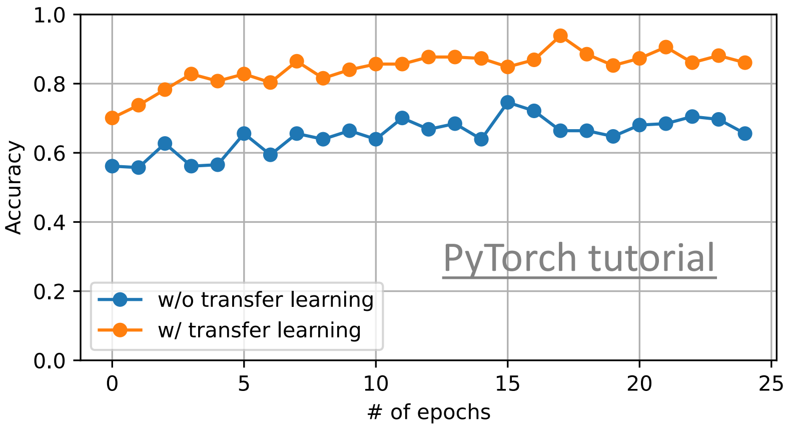
KEK, Computing Research Center

U Tokyo, Institute for AI and Beyond

Introduction

- **”Transfer learning”** technique has been successfully applied to many scientific field such as computer vision, natural language processing, etc





- Image classification: ”ants” vs “bees”
  - Significant improvement by transfer learning
  - Pre-trained on 1.2 million images with 1000 categories

Q: Is transfer learning technique beneficial for collider physics?

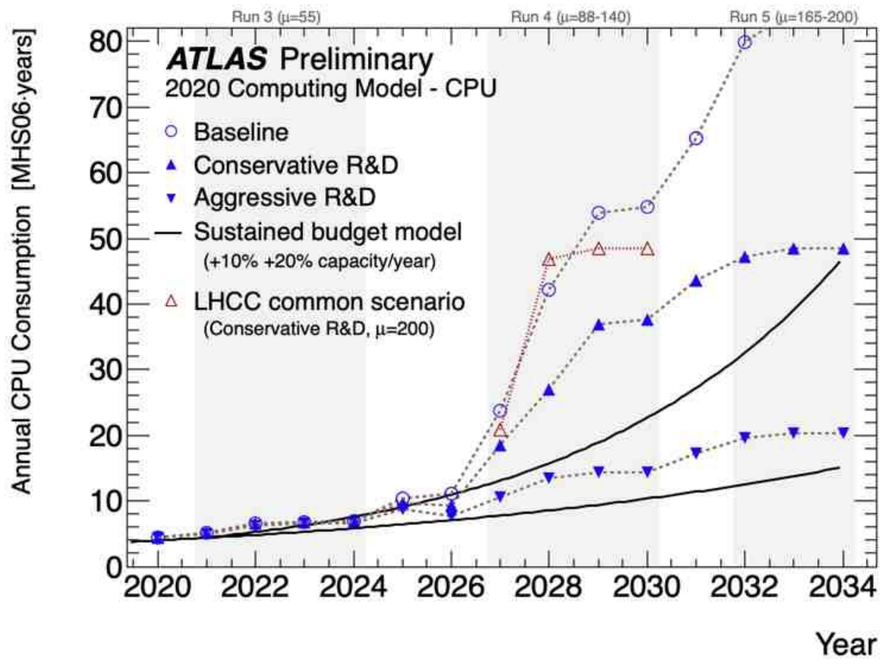
Sustainability

- Deep learning (DL) requires a large amount of data
  - Training data are typically generated by Monte Carlo (MC) simulations based on theories
    - However, MC simulations are computationally expensive
  - Electric power consumption, Green computing

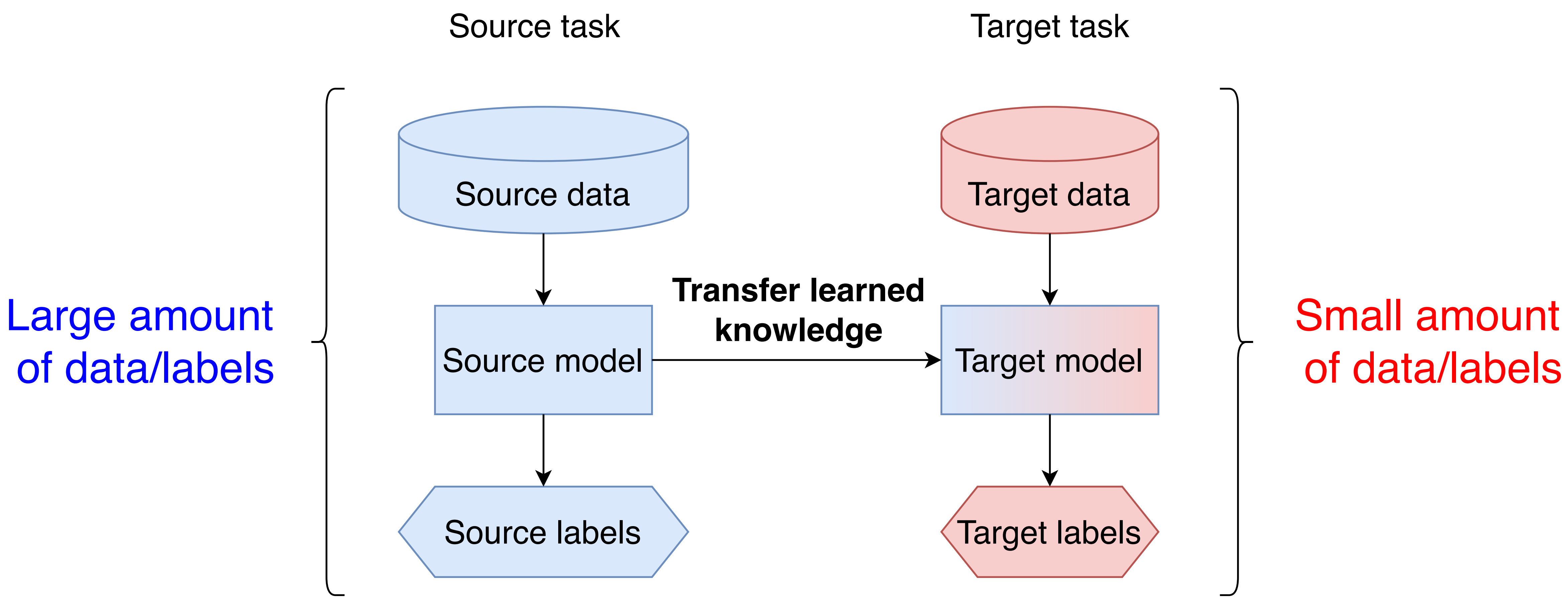
→ Maximizing DL performance with a limited number of data is a key concept

→ Transfer learning is a feasible approach

Expected CPU consumption (ATLAS)



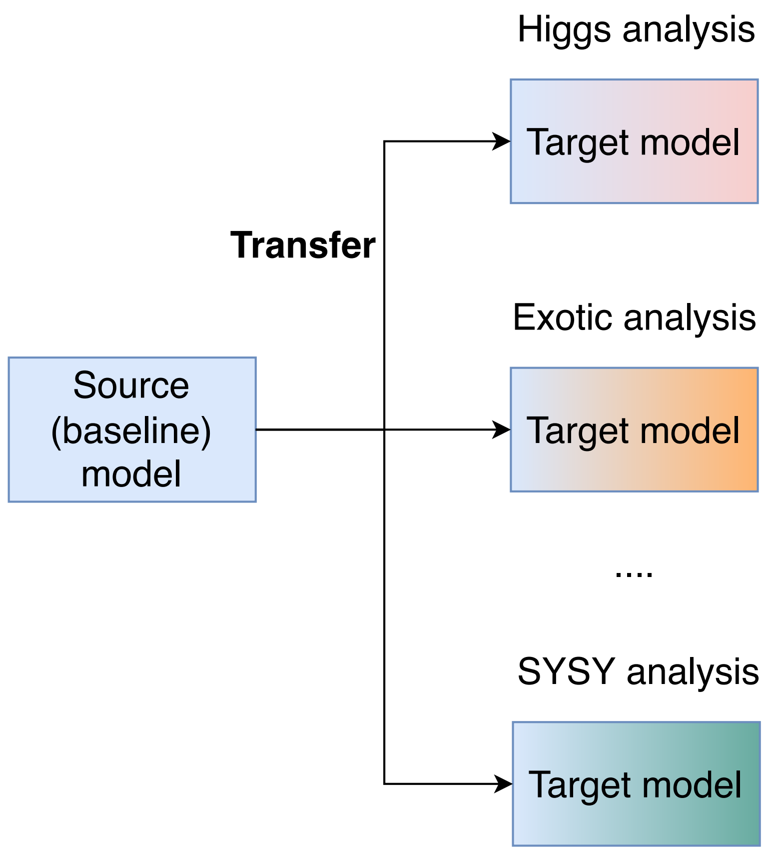
Transfer learning: basic idea



Use case of physics analysis

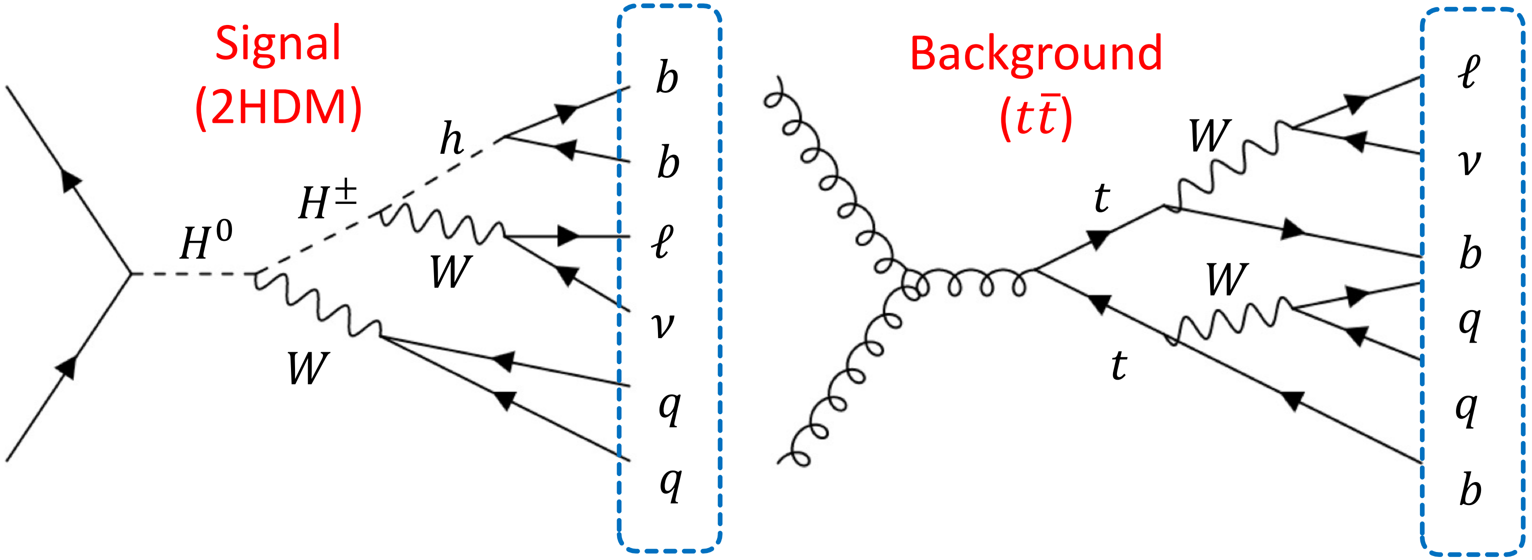
- There are many analysis channels in collider physics
  - Higgs, Exotic, SUSY analysis, etc
- Currently, dedicated DL models are trained from scratch for each analysis channel
  - Large amount of training data (MC data) for each channel

→ If transfer learning can be applied to different analysis channels, we can save computing resources (MC generation, training)



Physics processes

- To examine the transferability, several types of MC simulation data were generated by Madgraph + Pythia8 + Delphes
  - e.g.) $\boldsymbol{2HDM}$ vs $\boldsymbol{t\bar{t}}$



- Same final state particles ($l\nu bbjj$)
- **4-vector** $(\boldsymbol{p_T}, \boldsymbol{\eta}, \boldsymbol{\phi}, \boldsymbol{m})$ **+ object-type** for each object are inputs of DL models

→ 5 x 6 = 30 input variables in this example

Datasets

- Physics processes of **source** and **target** datasets:

Category	Bkg. vs Sig.	Signal mass	Final state	# of variables
Source dataset	$t\bar{t}$ vs $2HDM$	$H^0 = 425$ GeV, $H^{\pm}$=325 GeV	$lvbbjj$	5 x 6
Target dataset 1	$t\bar{t}$ vs $2HDM$	$H^0 = 500$ GeV, $H^{\pm}$=400 GeV	$lvbbjj$	5 x 6
Target dataset 2	$t\bar{t}$ vs Z'	$Z' = 1000$ GeV	$lvbbjj$	5 x 6
Target dataset 3	$ttbb$ vs ttH	Standard model	$lvbbbbjj$	5 x 8
Target dataset 4	Zvv vs $\tilde{g}\tilde{g}$	$\tilde{g} = 607$ GeV	$vvjjjj$	5 x 5

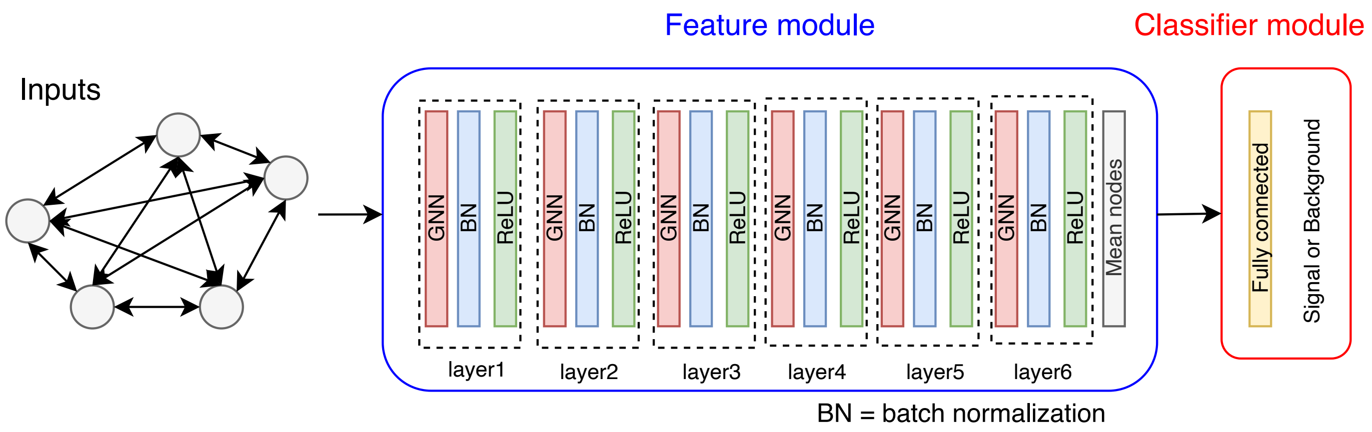
similar ↕ different

→ Simple expectation: transfer learning will work well for similar topology (physics)

Model overview

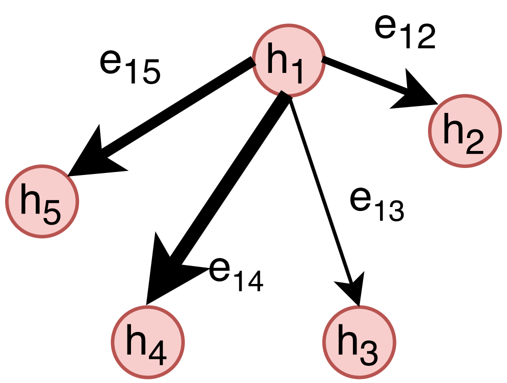
- To apply the transfer learning to various analysis channels, DL model must handle **variable number of objects** and be a **permutation invariant**

→ Graph Neural Networks (GNN)



- Model consists of two parts: **feature module** and **classifier module**
- Two types of GNN layer are examined:
  - w/ and w/o self-attention mechanism (**GATv2Conv** in DGL library)

Graph attention network



- Attention weights (edge features) represent importance of node (object) relations
  - e.g.) two b-jets from Higgs are important
  - Multi-head technique is used
  - Introduces small additional trainable parameters

- Number of trainable parameters:

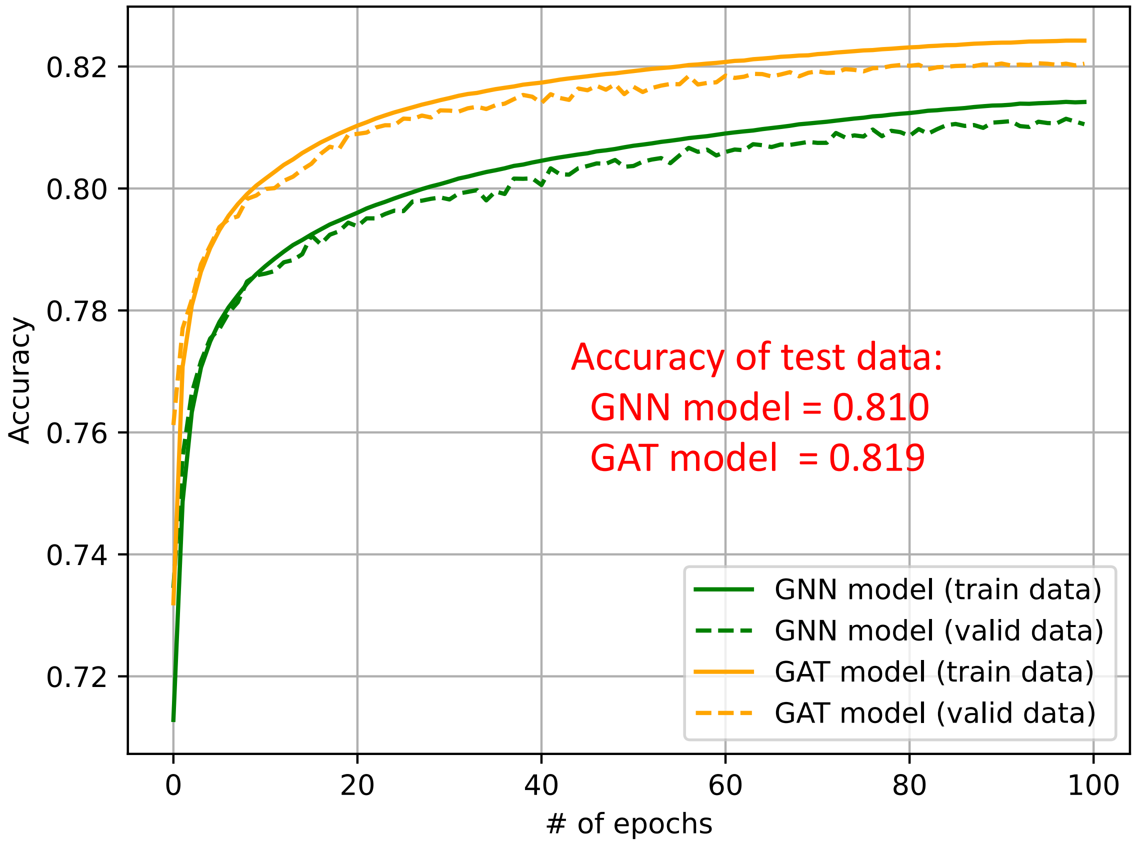
	Feature module	Classifier module	Total
w/o attention model (GNN model)	333312	514	333826
w/ attention model (GAT model)	334848 (↑1536)	514	335362 (↑1536)

"w/o attention model" performs simple message-passing (copying node features) without weights

Training of source task

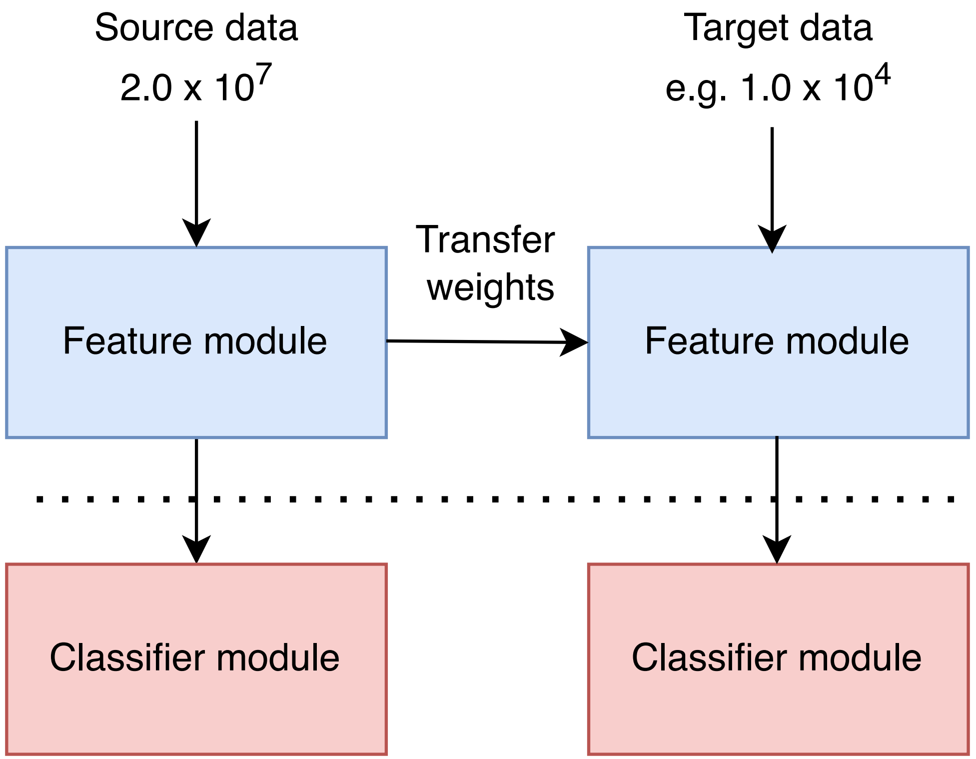
- Source task
  - Learning rate: CosineAnnealingLR
    - $1.0\times10^{-2} \sim 1.0\times10^{-4}$
  - Batch size: 2048, # of epochs: 100
  - Grid search for model architecture:
    - # of layers: [5, **6***, 7, 8]
    - # of hidden features: [128, **256***, 512, 1024]
    - # of multi-heads: [2, **4***, 8, 16]

*Bold parameters are selected



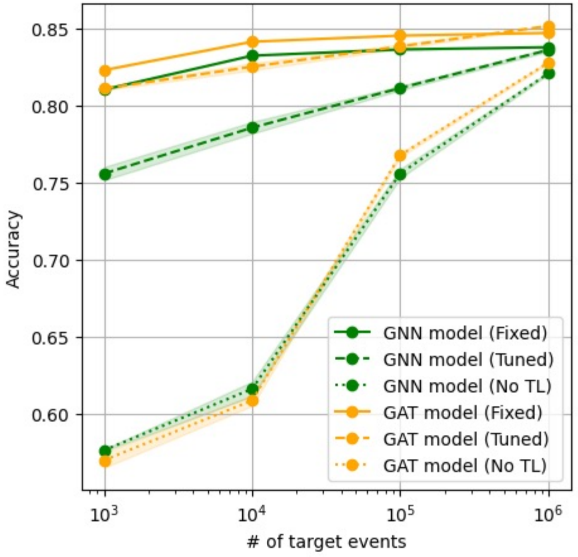
Training of target task

- Target tasks
  - **Only weights of the feature module are transferred from source task to target task**
    - **Fixed:** the transferred weights are not updated during the training with target datasets
    - **Tuned (fine-tuning):** the transferred weights are updated (tuned) during the training with target datasets
  - **Classifier module is trained from scratch**
  - Same learning rate with the source task
  - Batch size: 256, # of epochs: 100
  - Cases of 10^3, 10^4, 10^5, 10^6 target events are examined

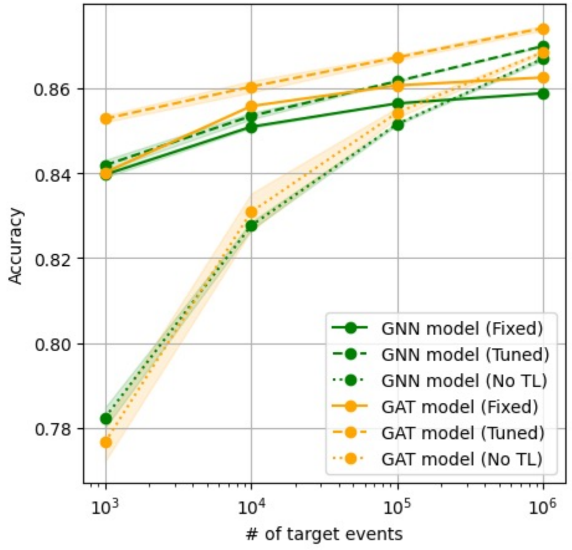


Result: accuracy

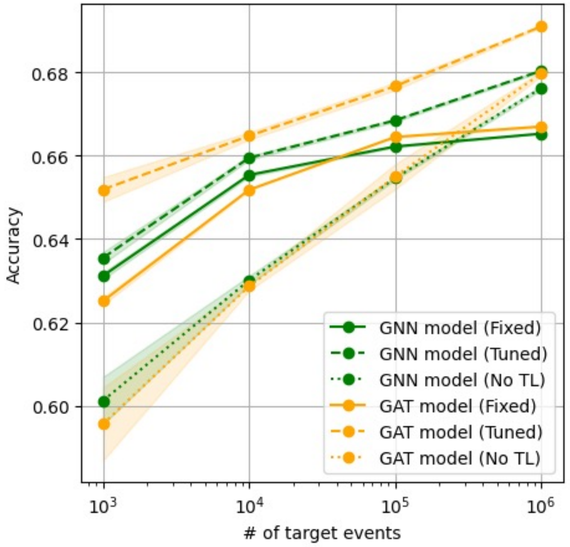
Target dataset 1
($t\bar{t}$ vs $2HDM$)



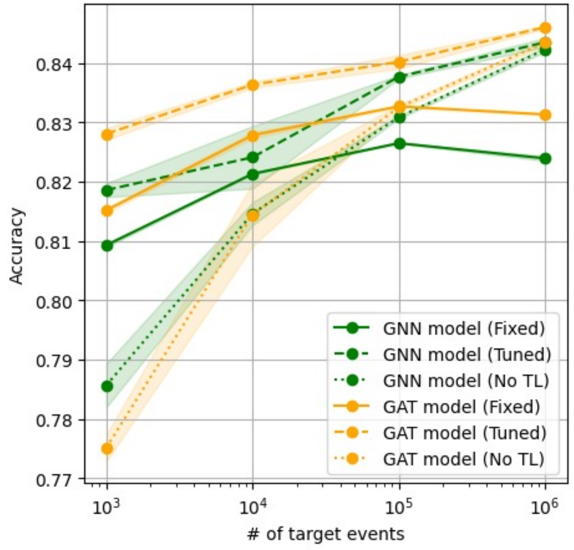
Target dataset 2
($t\bar{t}$ vs Z')



Target dataset 3
($ttbb$ vs ttH)



Target dataset 4
($Z\nu\nu$ vs $\tilde{g}\tilde{g}$)



- Significant improvement if topology is similar
- Fixed weights decreases performance if # of events are sufficient

Summary

- Transfer learning technique is applied to the event classification in collider physics
  - Graph neural network architecture allow us to adapt different analysis channels
  - Transfer learning provides a significant improvement when target dataset is insufficient
    - E.g.) ~20% improvement (target dataset1) for target 1.0×10^4 events
    - Fine-tuning is effective in absorbing topology differences
    - Similar performance between w/ and w/o transfer learning for target > 1.0×10^6 events

Backup

Technical details

- DL models are implemented using PyTorch + DGL libraries
  - Git link to source codes
- Generated MC simulation events:
  - 2~3 days to generate source and target datasets with ~300 CPU cores
  - Difficult to increase statistics more

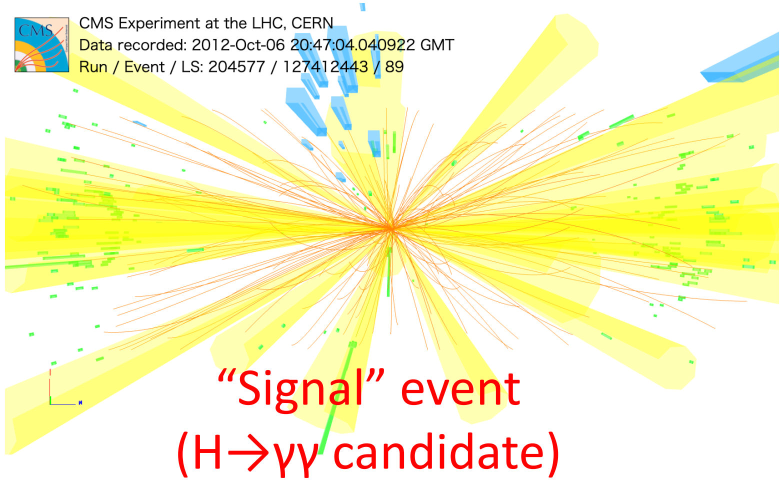
	Train data	Valid data	Test data
Source dataset	2.0×10^7	1.0×10^5	1.0×10^5
Target dataset for each	1.0×10^6	1.0×10^5	1.0×10^5

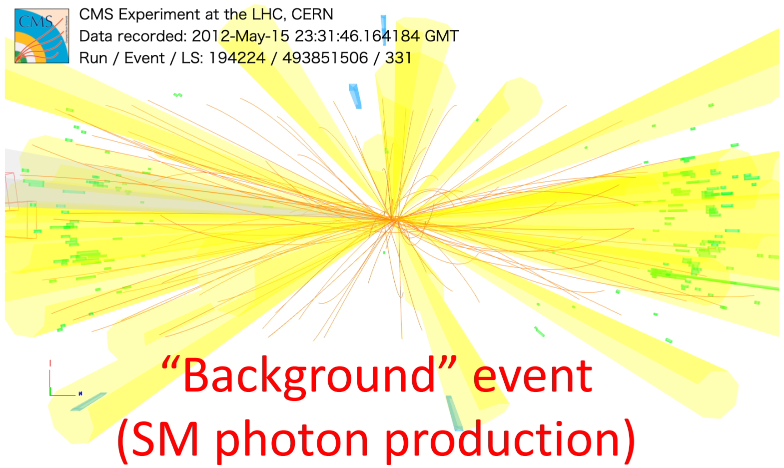
- GPU architecture for DL training
  - Nvidia A100 x 1, ~7 hours for training of source task (100 epochs)

Event classification

- **"Event classification"** is a typical problem in collider physics
  - Interesting signal events are separated from background events
  - Based on the information of reconstructed particles (objects), lepton, jets, missing E_T, etc

CMS event display



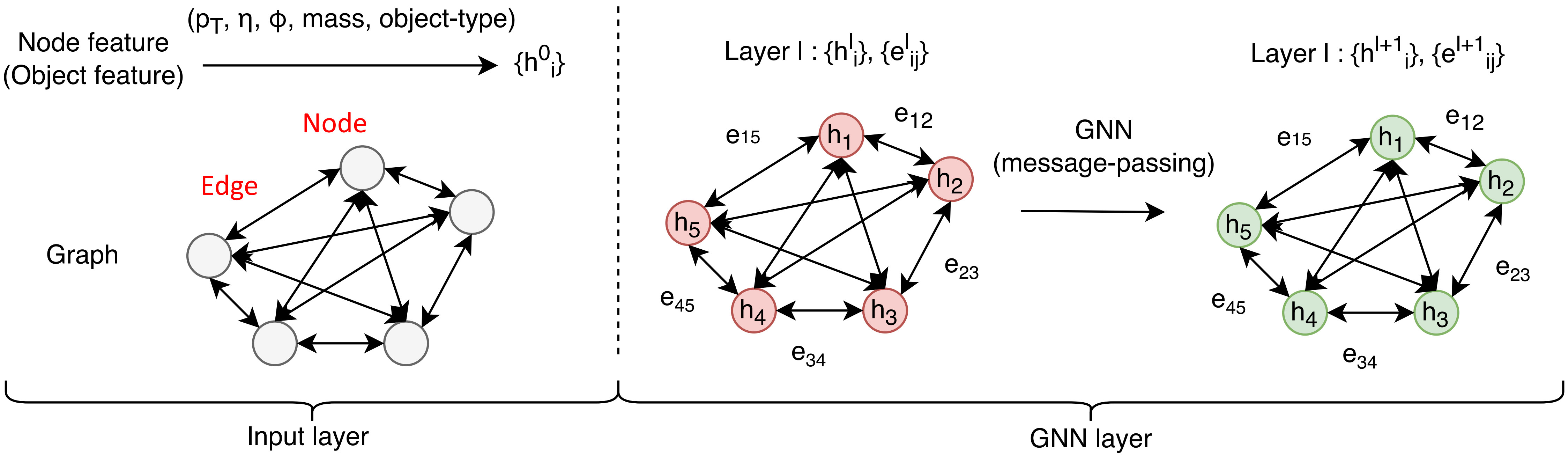


→ There are many studies using Deep learning for this event classification problem

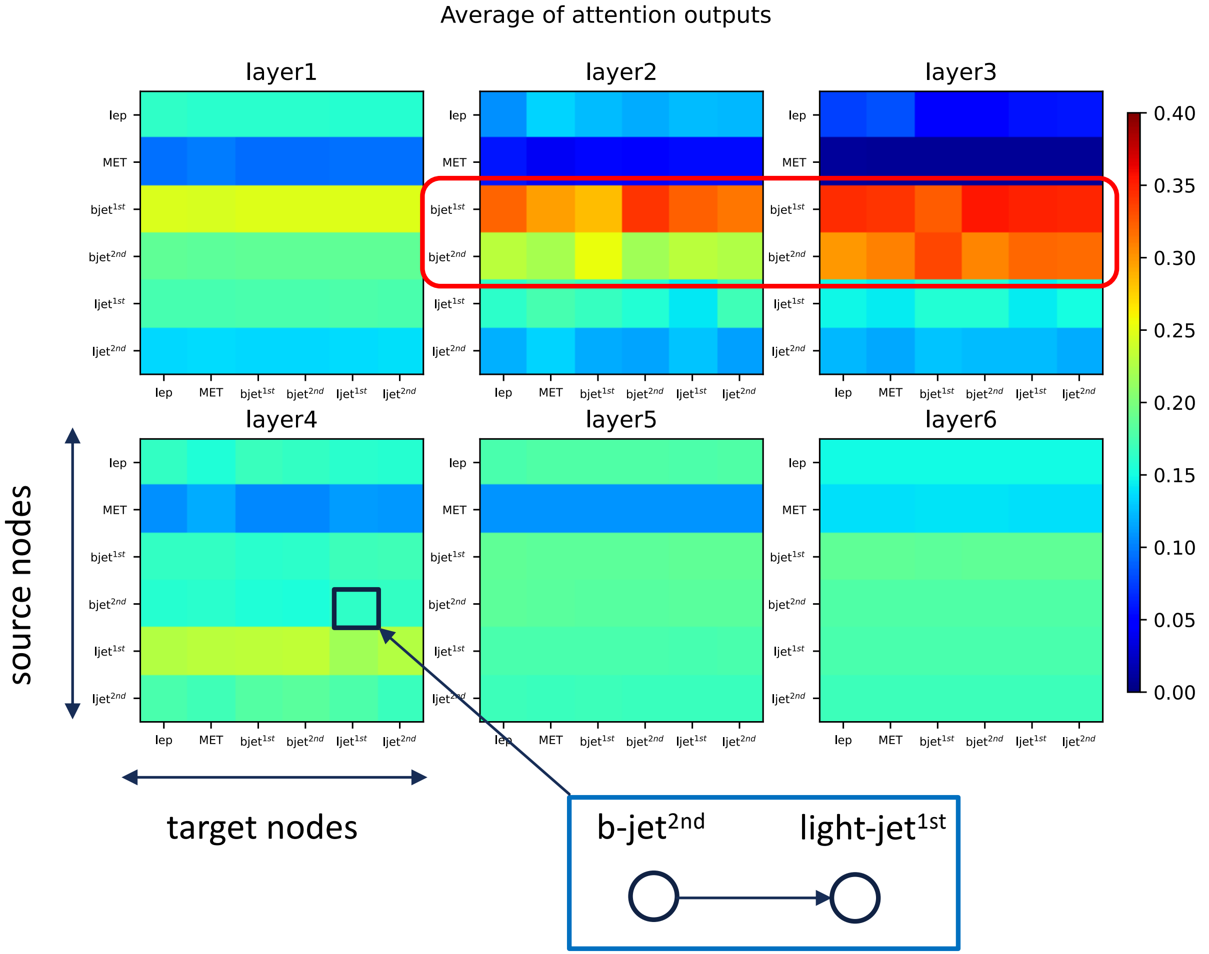
DL model

- To apply the transfer learning to various analysis channels, DL model must handle **variable number of objects** and be a **permutation invariant**

→ Graph Neural Networks (GNN)



Attention outputs: source dataset



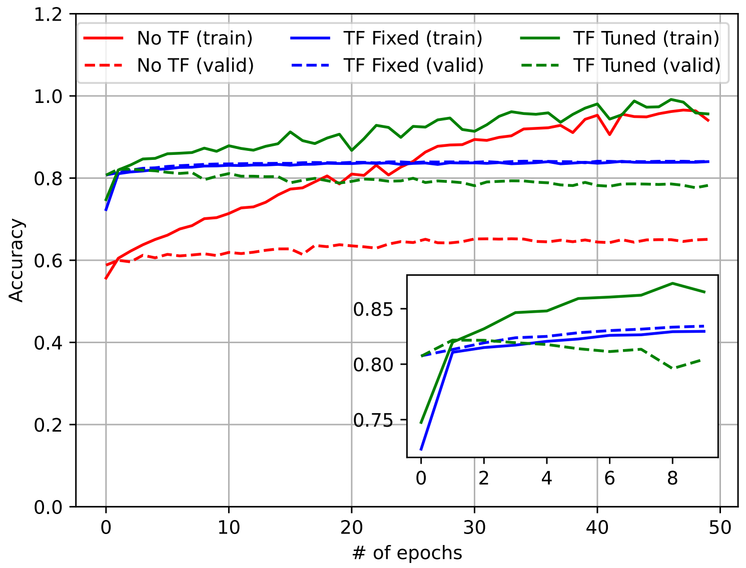
$$\text{Avg. of attn. outputs} = \frac{1}{\text{nevnets} \times \text{mheads}} \sum_{i=1}^{\text{nevents}} \sum_{j=1}^{\text{mheads}} (\text{attn. outputs}^{i,j})$$

- Jets from b-quark (b-jets) are considered important in GAT model
  - Higher values of attention outputs
  - Consistent with our knowledge: H→bb is a discriminant signature

Result: over-training

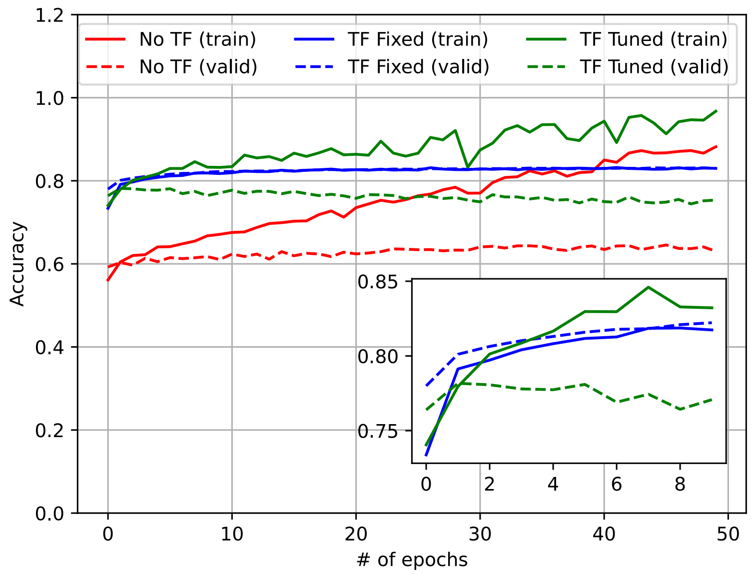
- Example of results
  - Target dataset 1 ($t\bar{t}$ vs $2HDM$)
  - 1.0 x 10^4 target events

GAT model



~20%

GNN model

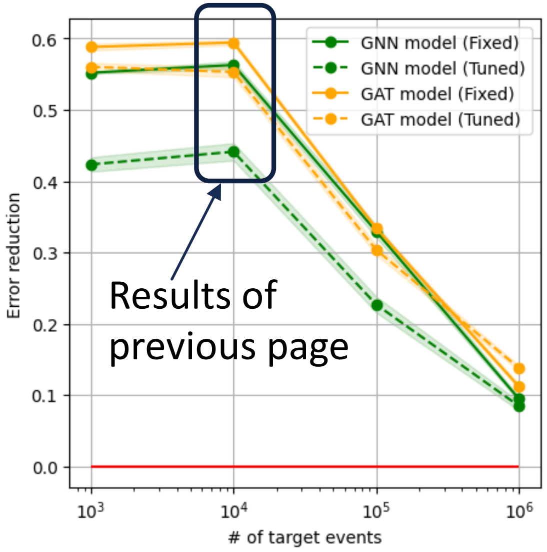


~20%

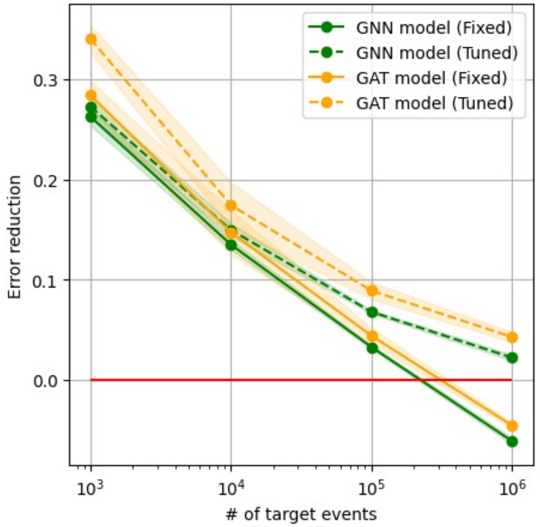
- ~20% improvements by the transfer learning (TL) in these examples
  - TL with tuned weights still causes over-training if target dataset is small
  - Fixed weights show better performance in these cases

Result: error reduction

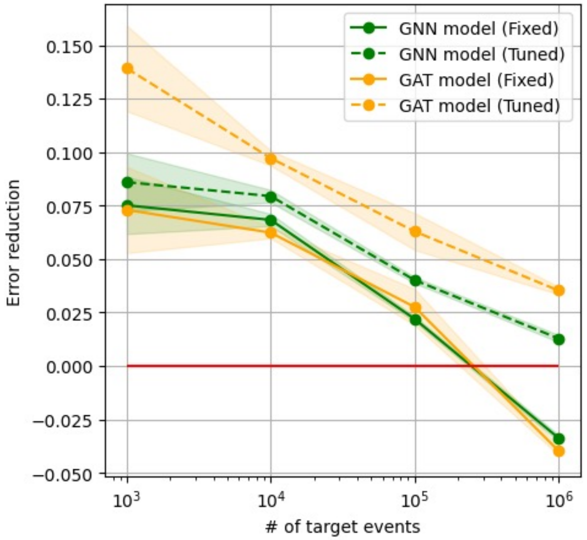
Target dataset 1
($t\bar{t}$ vs $2HDM$)



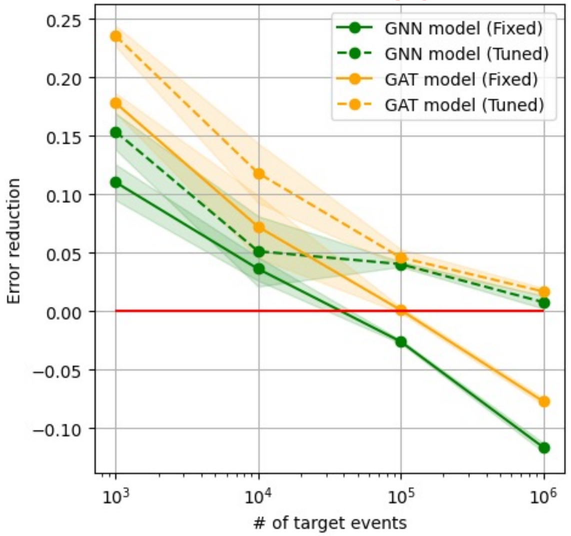
Target dataset 2
($t\bar{t}$ vs Z')



Target dataset 3
($ttbb$ vs ttH)



Target dataset 4
($Z\nu\nu$ vs $\tilde{g}\tilde{g}$)



$$\text{Error Reduction} = 1 - \frac{\text{Error}^{\text{TF}}}{\text{Error}^{\text{No-TF}}} \begin{cases} > 0 & \text{improvement by TF} \\ \leq 0 & \text{no improvement by TF} \end{cases}$$

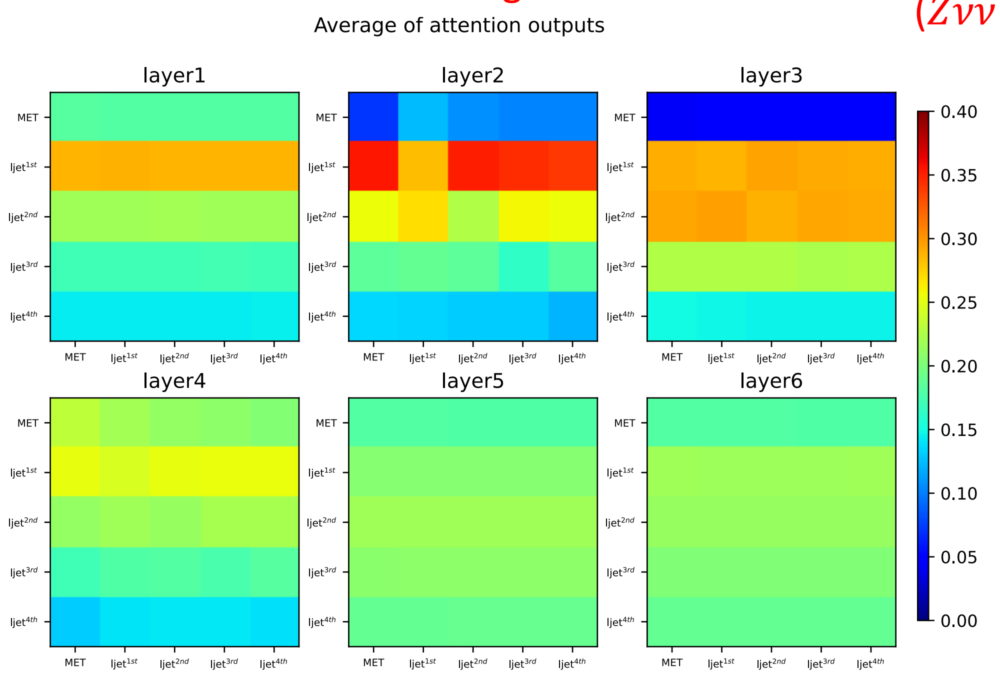
$$\text{Error} = 1 - \text{Accuracy}$$

- Significant improvement if topology is similar
- Fixed weights decreases performance if # of events are sufficient

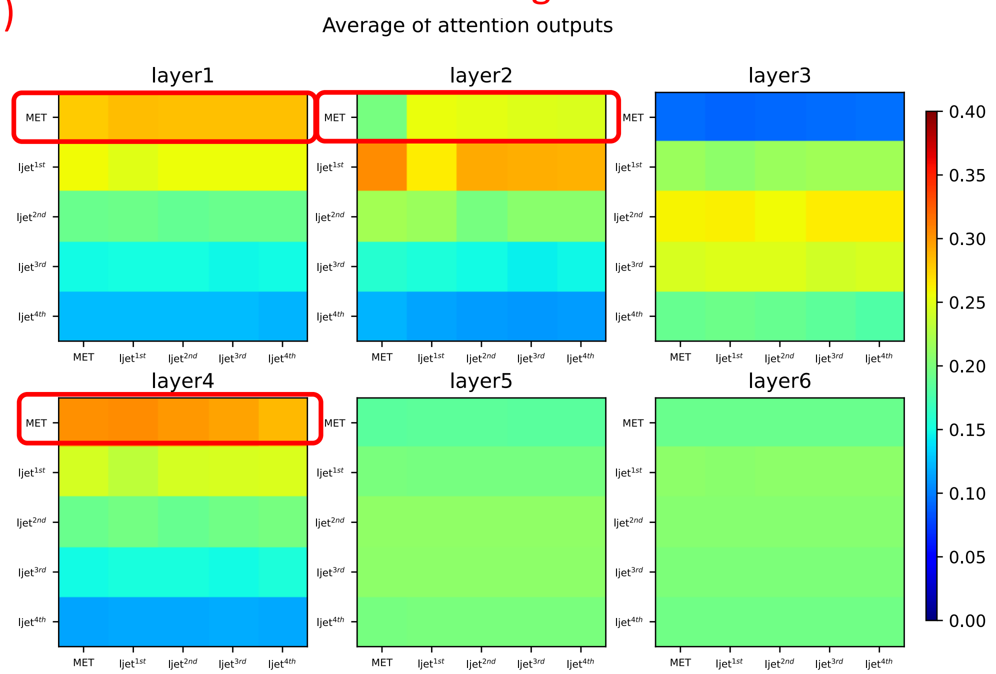
Attention outputs: target dataset

Target dataset 4
($Z\nu\nu$ vs $\tilde{g}\tilde{g}$)

Fixed weights



Tuned weights



- Fine-tuning increases importance of missing energy (MET)
  - Effective in absorbing differences between source and target topologies