Study of model construction and the learning for hierarchical models

Learning to Discover: AI and High Energy Physics conference

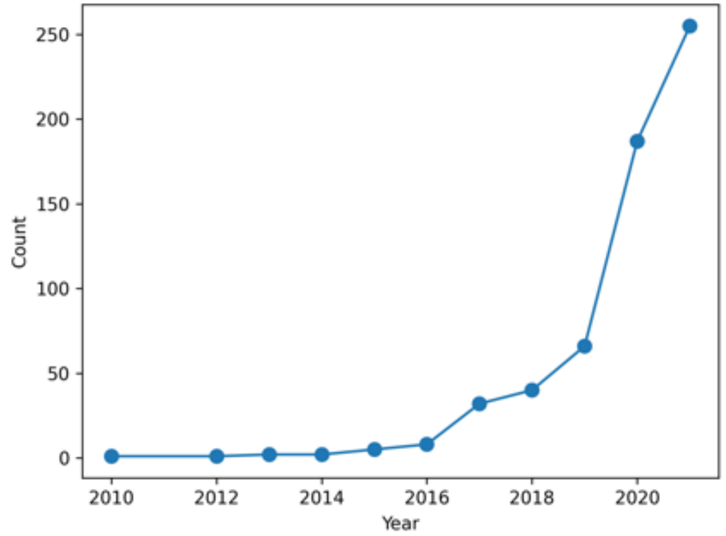
27 / 04 / 2022

ICEPP[A], KEK[B], Beyond AI[C]

Masahiko Saito[AC], Tomoe Kishimoto[BC], Masahiro Morinaga[AC], Sanmay Ganguly[AC], Junichi Tanaka[AC]

Motivation

of papers at HEPML-LivingReview

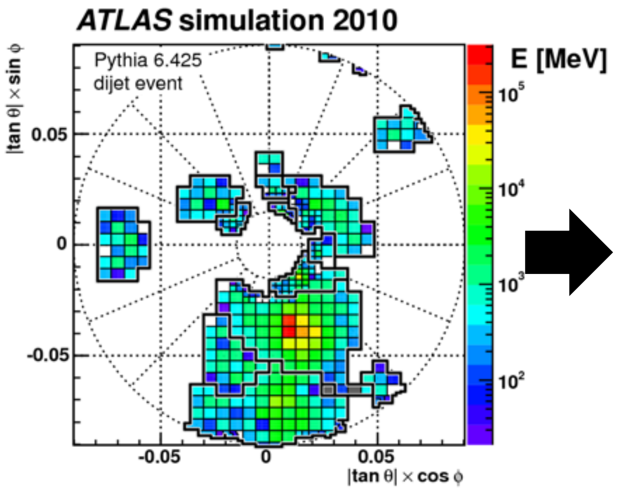


- The application of deep learning in the HEP field is growing.
  - Focusing on a single task (Event classification, PID, …)

- Most of problems consist of **multiple small tasks**.

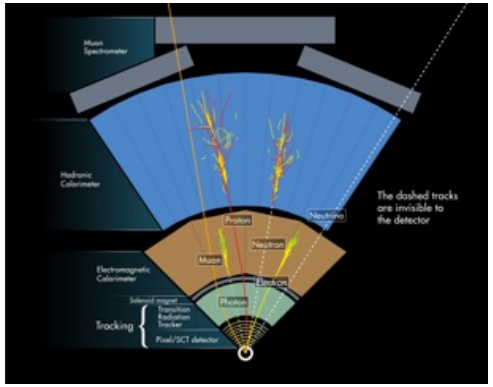
Step from raw data to physics analysis

Clustering/Tracking



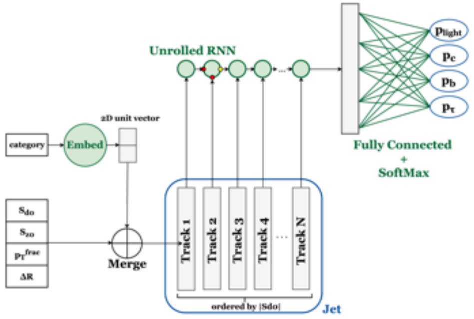
Eur. Phys. J. C 77 (2017) 490

Physics object reconstruction



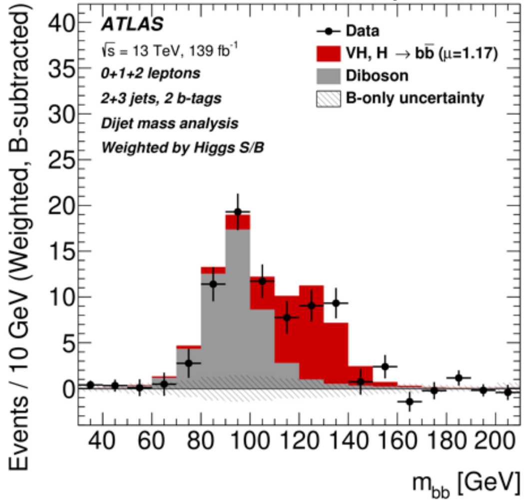
CERN-EX-1301009

Particle identification



ATL-PHYS-PUB-2017-003

Statistical analysis



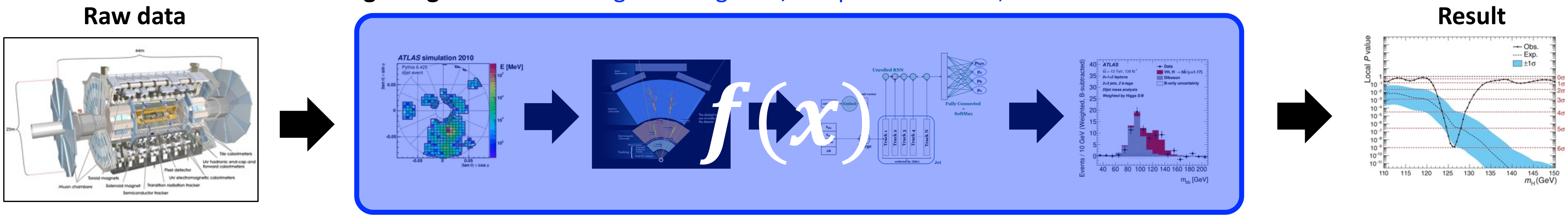
Eur. Phys. J. C 81 (2021) 178

Motivation

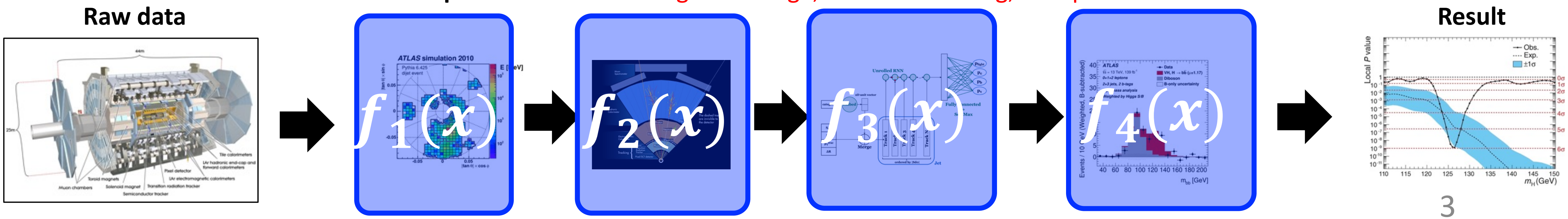
- The application of deep learning in the HEP field is growing.
  - Focusing on a single task (Event classification, PID, …)

- Most of problems consist of **multiple small tasks**.

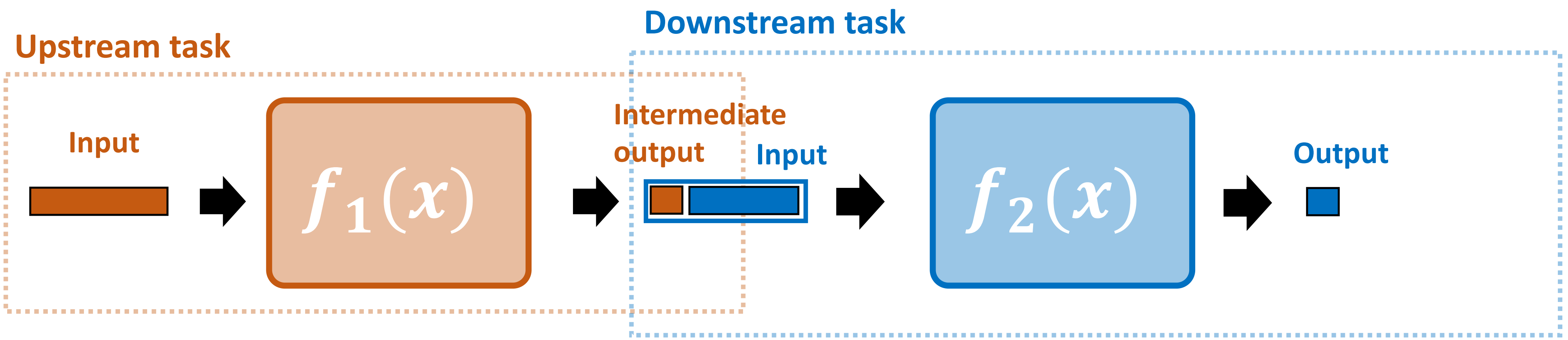
Large single DL model: Huge training data/compute resources, blackbox



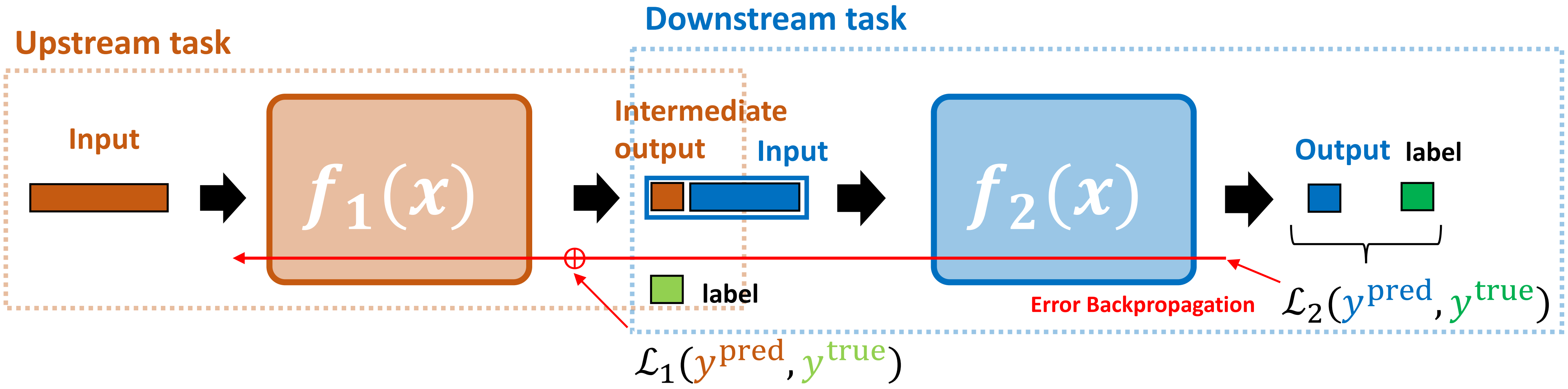
multi-step DL model: Reflecting knowledge, efficient learning, interpretable



Multi-step deep learning model



Multi-step deep learning model



Training with additional label for intermediate output means more injection of our knowledge.

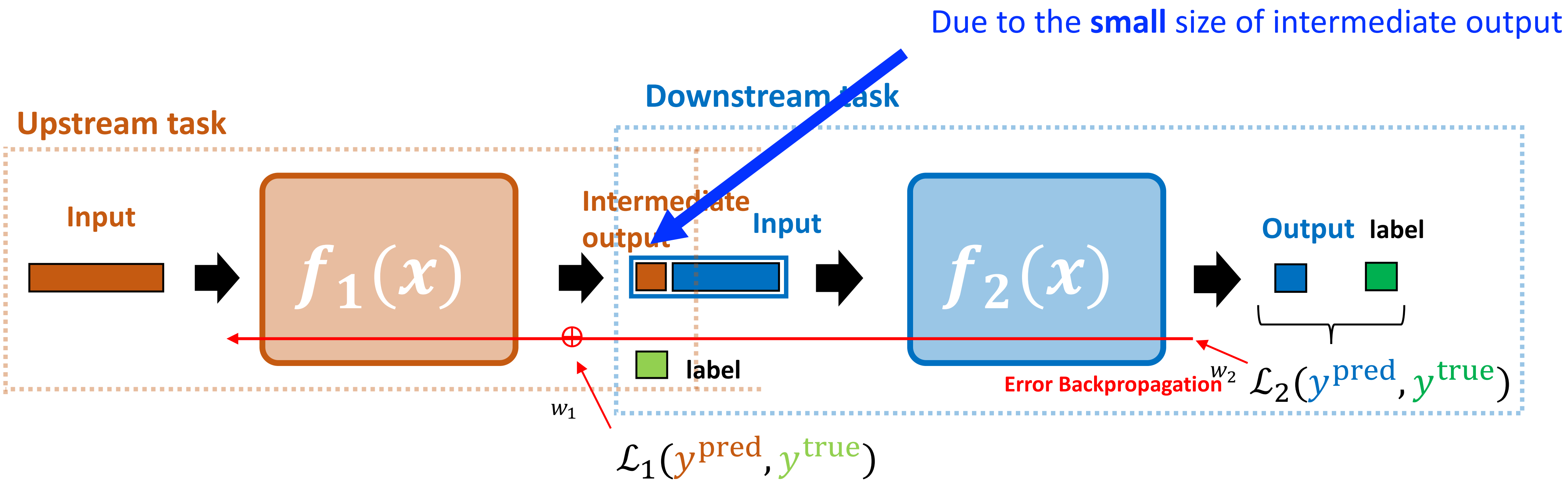
- We train the multi-step DL model via weighted sum of each task's loss

$$\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2$$

- Issues
  1. Limited representation power due to the shape of the intermediate output and loss function
  2. Necessary to tune loss coefficients (w_1, w_2) for each task as hyperparameters

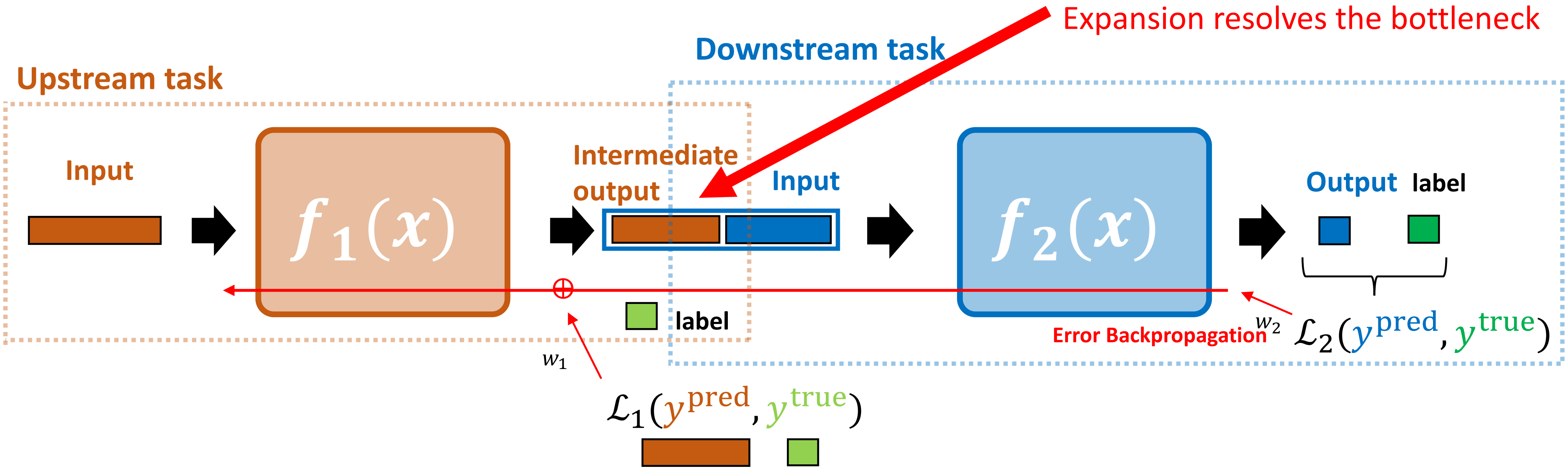
Solution

- Issues
  1. Limited representation power due to the shape of the intermediate output and loss function
  2. Necessary to tune loss coefficients (w_1, w_2) for each task as hyperparameters



Solution

- Issues
  1. Limited representation power due to the shape of the intermediate output and loss function
  2. Necessary to tune loss coefficients (w_1, w_2) for each task as hyperparameters

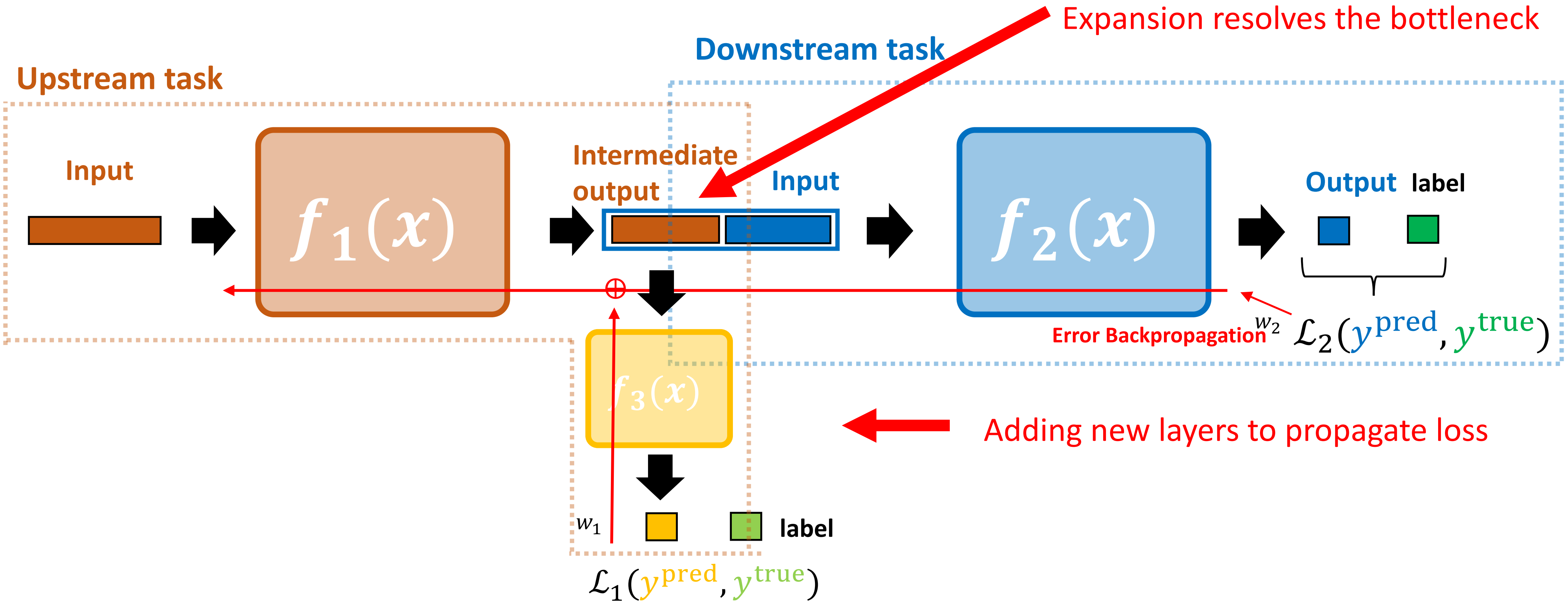


We cannot use the upstream task's loss function

due to a mismatch of intermediate output and label.

Solution

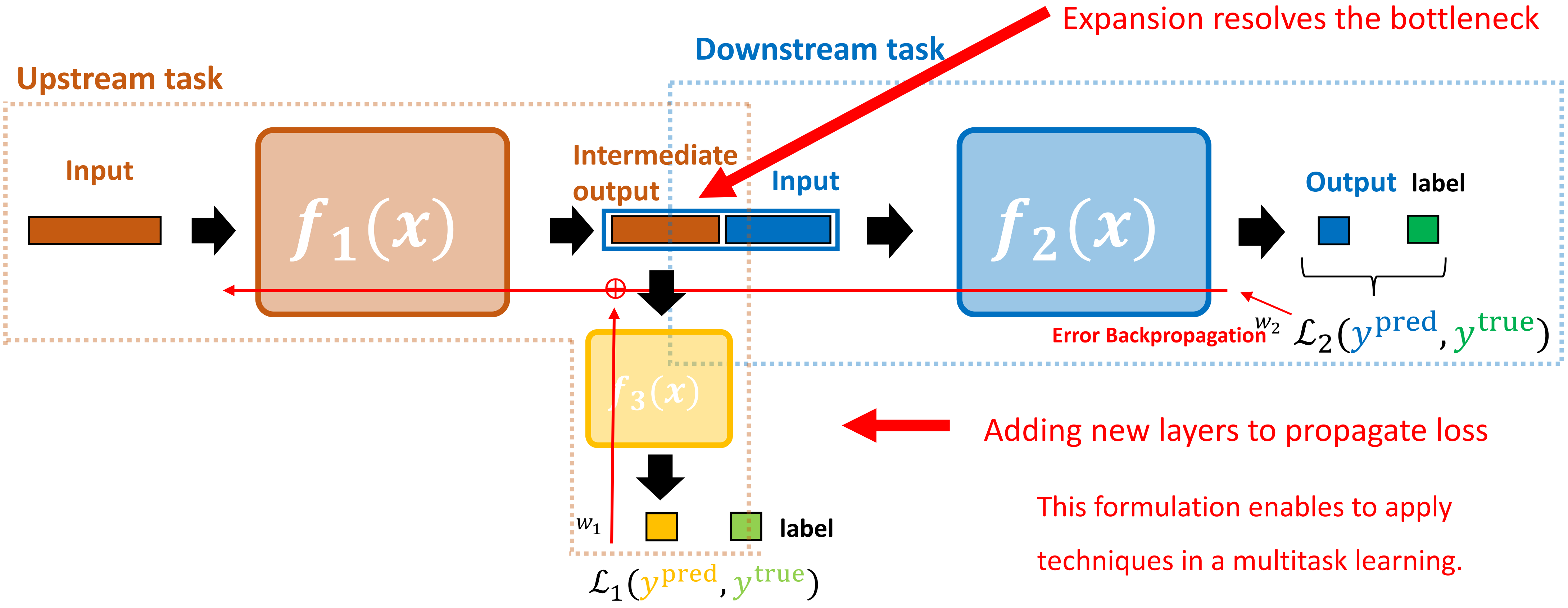
- Issues
  1. Limited representation power due to the shape of the intermediate output and loss function
  2. Necessary to tune loss coefficients (w_1, w_2) for each task as hyperparameters



Solution

- Issues

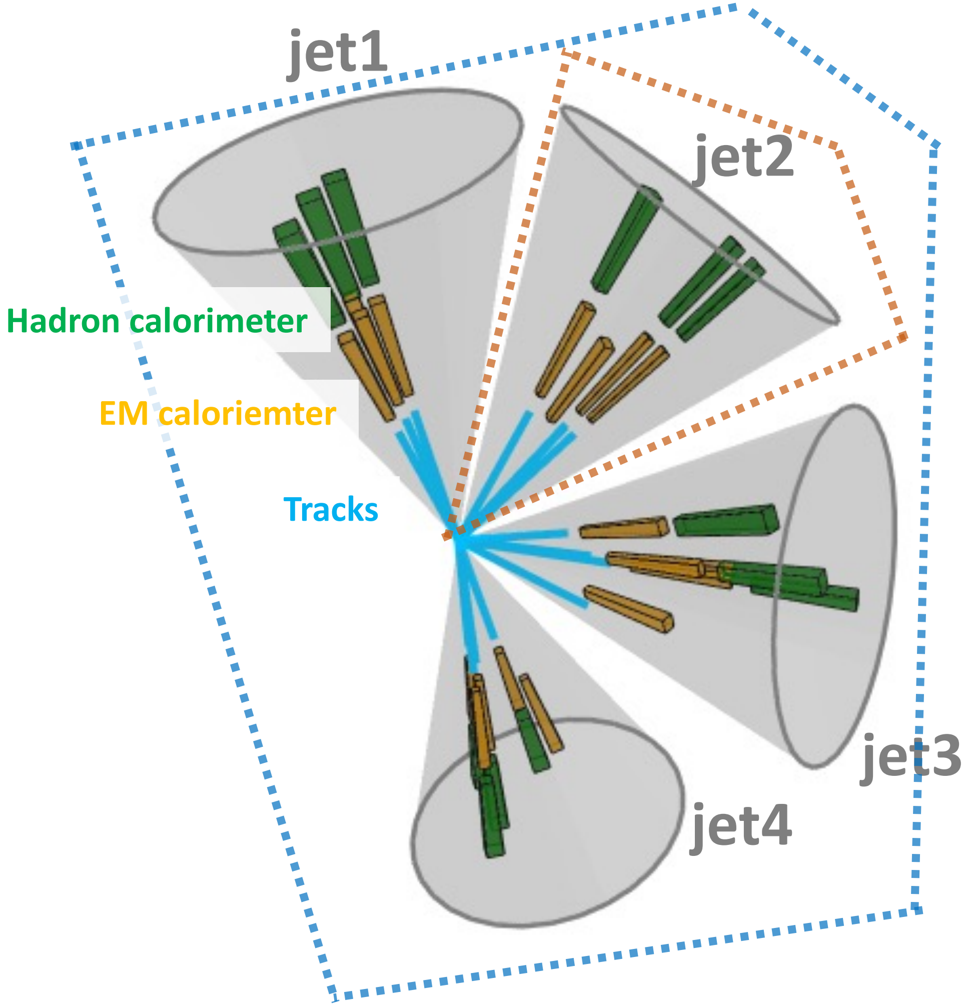
1. Limited representation power due to the shape of the intermediate output and loss function
2. Necessary to tune loss coefficients (w_1, w_2) for each task as hyperparameters



This formulation enables to apply techniques in a multitask learning.

➡ Applied to multi-step tasks: “Tau identification” and “Classification of $H \rightarrow \tau\tau$ / $Z \rightarrow \tau\tau$”

Application in HEP: Classification of $\mathrm{H} \to \tau\tau$ / $\mathrm{Z} \to \tau\tau$



Upstream task: Tau ID (classification of τ-jet / light-jet)

Input: momentum vector of jet constituents (max. 50 constituents)

Output: Probability that a jet's origin is a tau particle

Downstream task: Event classification (classification of H / Z)

Input: Jets (max. 8) features in events

- four-vector
- output of the upstream task

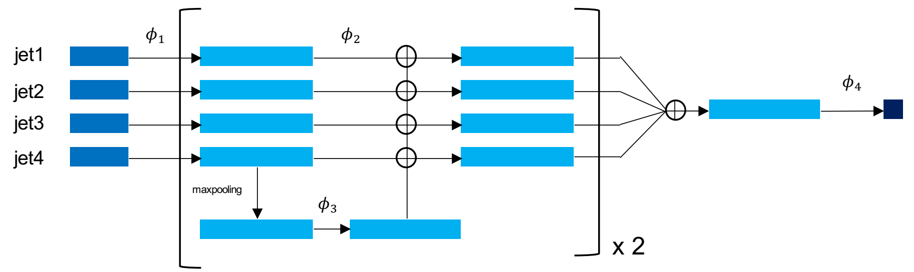
Output: Probability that the event contains Higgs boson

Dataset

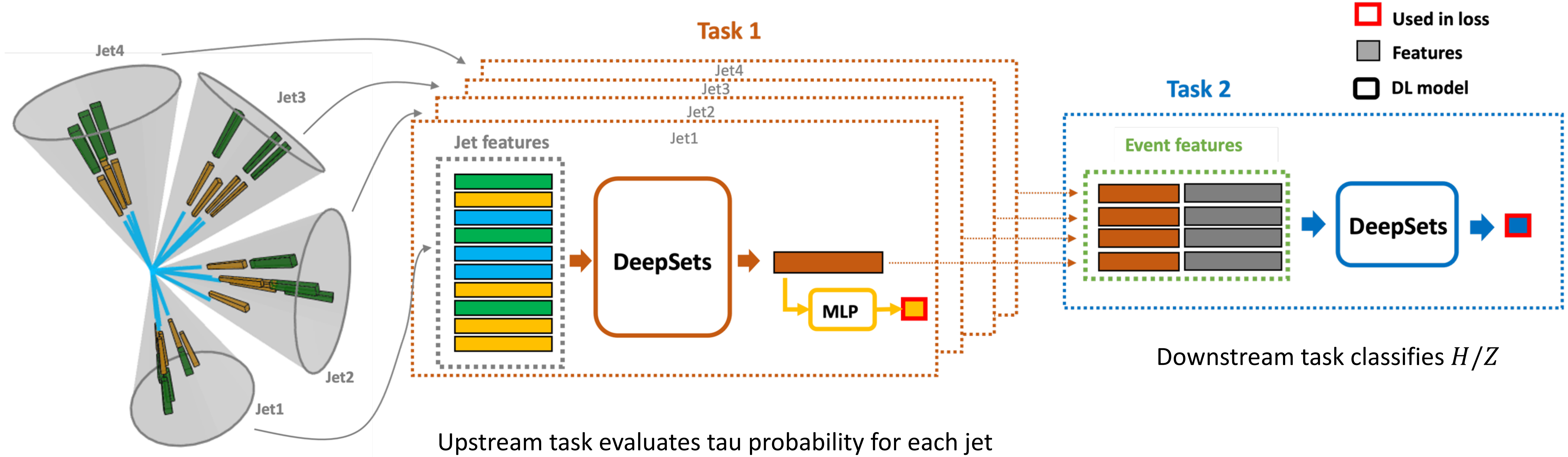
- Simulated data (Pythia8 + Delphes)
- $\langle \mu \rangle$ = 50
- Only hadronically decaying tau (tau-jet)
- 100k events for $H \to \tau\tau,\ Z \to \tau\tau$

Deep learning model

- DeepSets are used for both tasks
  - DeepSets can handle a variable length of inputs
- Adam (lr = 0.001) with early stopping (max patients = 10)
- Use cross-entropy loss for both tasks



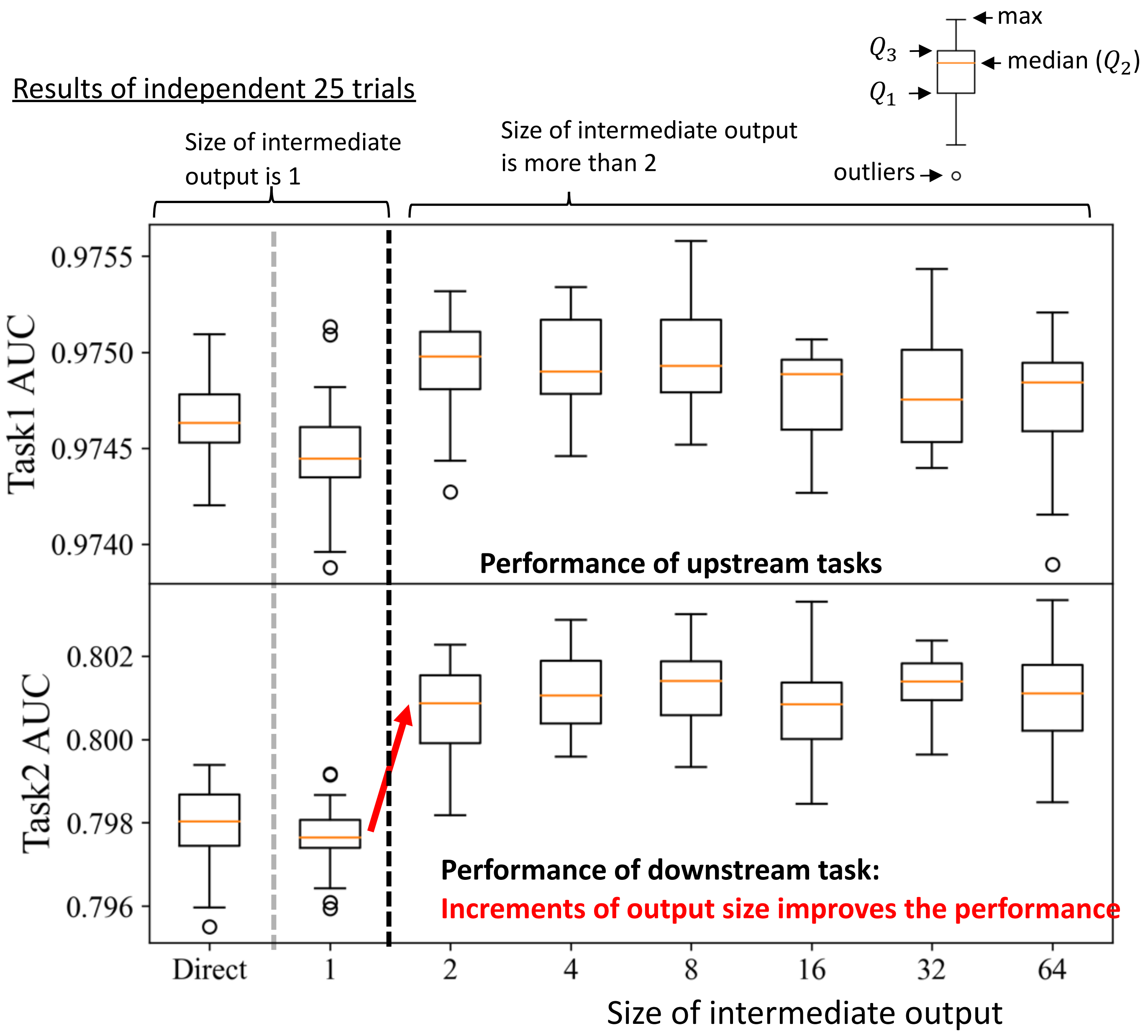
DeepSets



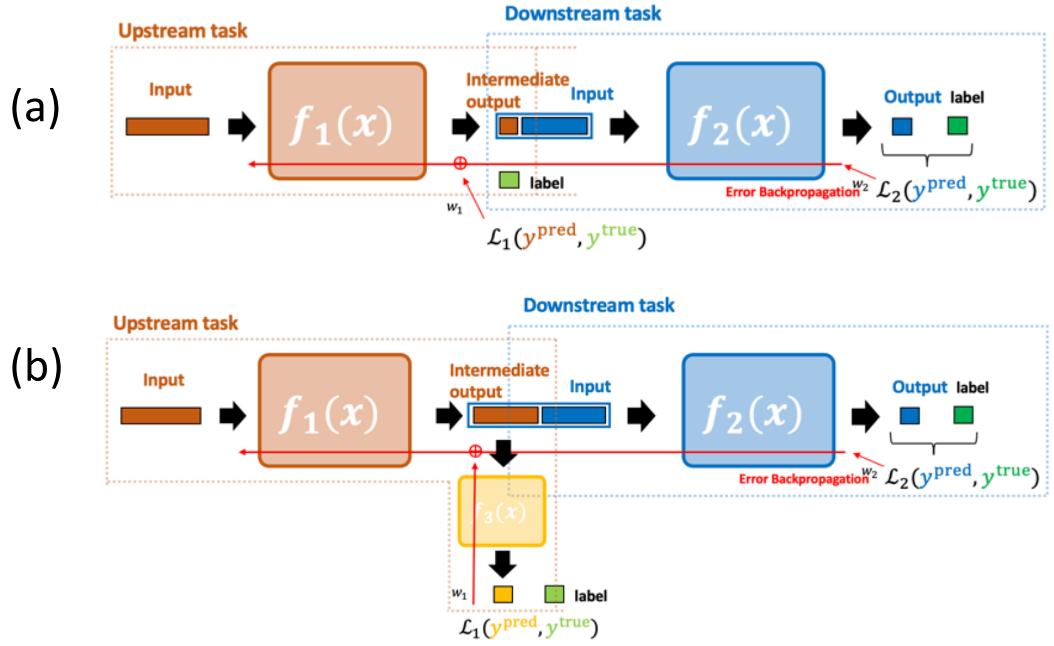
Upstream task evaluates tau probability for each jet

Downstream task classifies H/Z

Dependency of intermediate output size

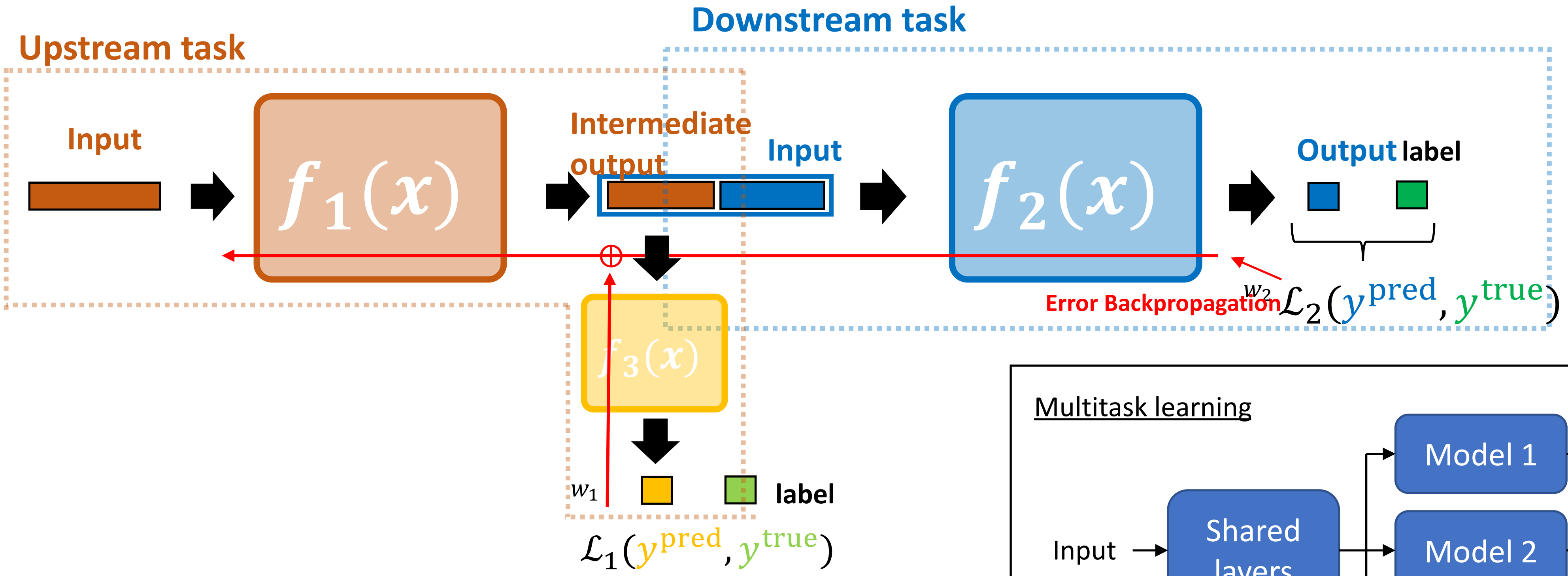


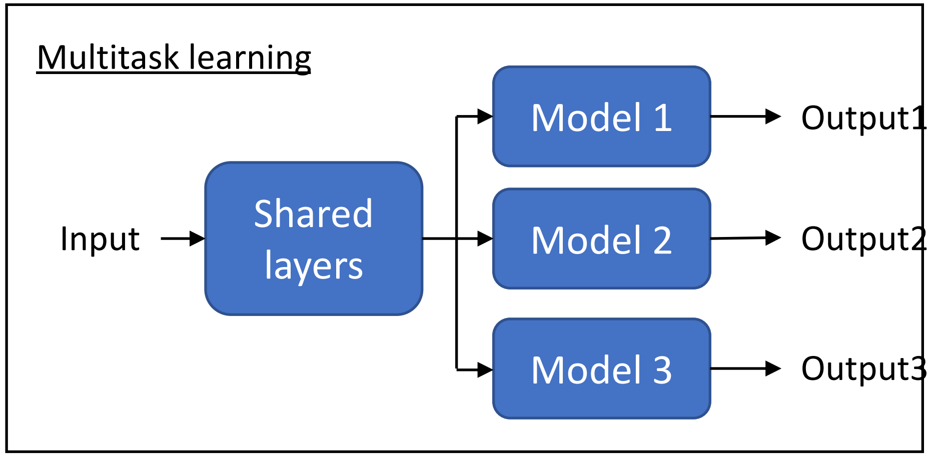
(a) Direct connection (b) Training with additional NN



- Bottleneck of information due to a direct use of upstream output as the input of downstream task
- Adequate size of intermediate output is important for the downstream task's performance.
  - Learning something useful other than tau prob
  - Simultaneous training of multiple models contributes to improve the performance

Adaptive optimization of loss coefficient



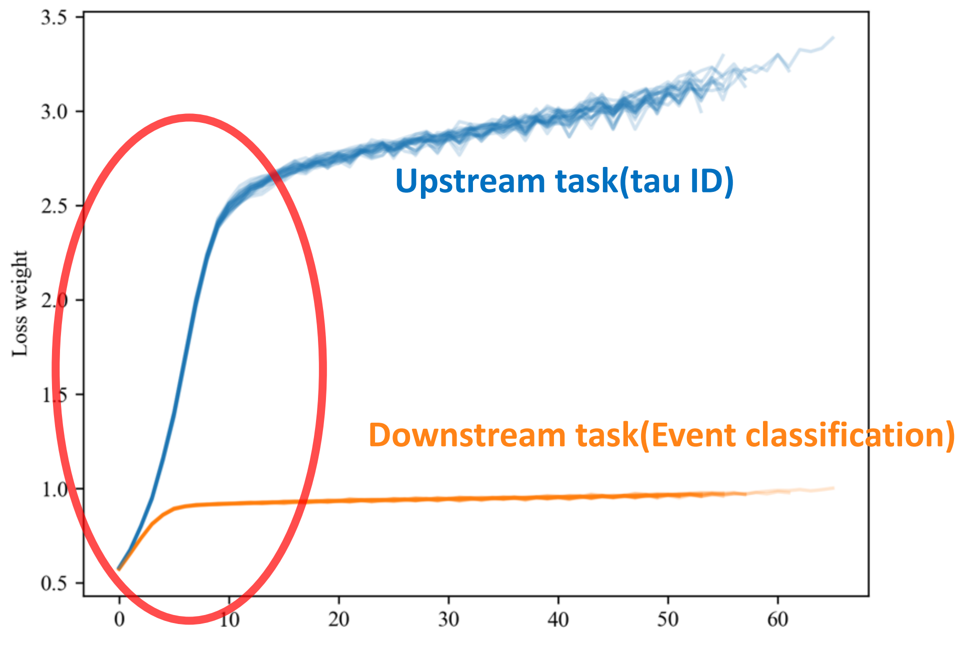


- Possible to regard it as a kind of **multitask learning**
  - Intermediate output is regarded as shared features
- There are some methods proposed to efficiently learn loss of $\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2$ in the context of multitask learning
- Uncertainty weighting
  - Target function: $\mathcal{L} = \sum_i \left(\frac{1}{2\sigma_i^2}\mathcal{L}_i + \log\sigma_i\right)$ (σ_i is a trainable parameter, not NN outputs)
  - Loss coefficients are tuned depending on each loss absolute value.

Adaptive optimization of loss coefficient

$$\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2$$

Coefficients are automatically tuned



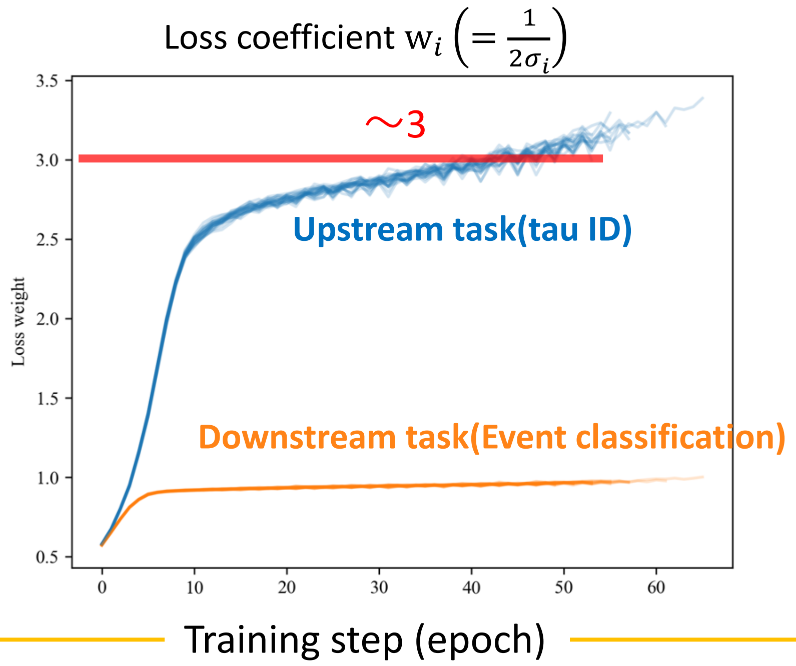
Algorithm	Upstream task AUC	Downstream task AUC
$(w_1, w_2) = (0.1, 1.0)$	0.9689	0.7993
$(w_1, w_2) = (1.0, 1.0)$	0.9748	**0.8013**
$(w_1, w_2) = (10, 1.0)$	**0.9753**	0.8005
Uncertainty weighting	**0.9753**	**0.8015**

w_i is fixed in training.
Necessary to tune HPs depending on $\mathcal{L}_i$

→ Good performance without tuning of w_i

Adaptive optimization of loss coefficient

$$\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2$$



Upstream task: **Cross-entropy**

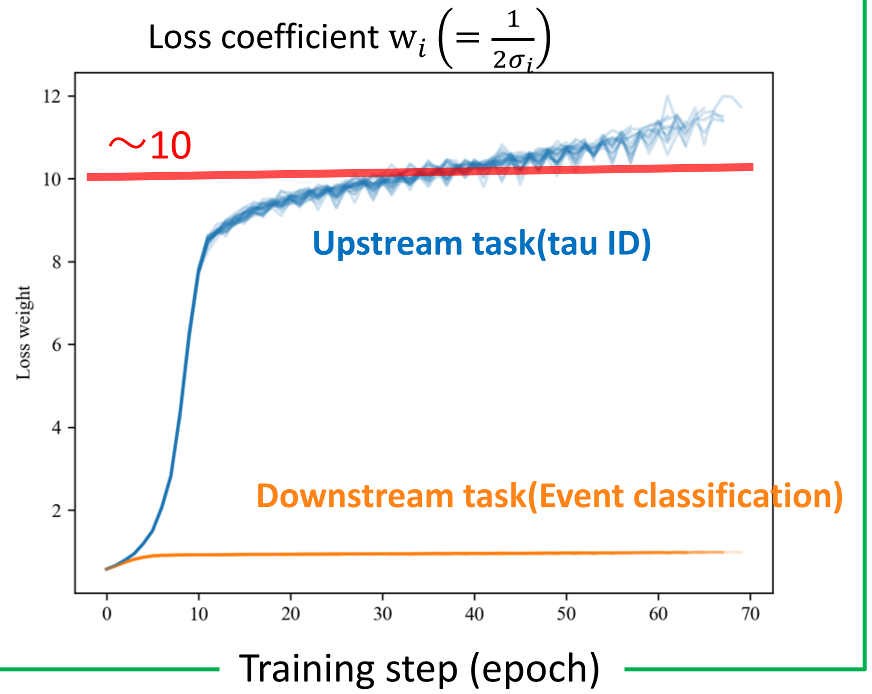
Downstream task: Cross-entropy

Different scaled loss function

(Occurs when using both classification and regression)

Upstream task: **Mean squared error**

Downstream task: Cross-entropy



	(Cross-entropy, Cross-entropy)		(Mean squared error, Cross-entropy)	
Algorithm	Upstream task AUC	Downstream task AUC	Upstream task AUC	Downstream task AUC
$(w_1, w_2) = (0.1, 1.0)$	0.9689	0.7993	0.9618	0.7985
$(w_1, w_2) = (1.0, 1.0)$	0.9748	**0.8013**	0.9717	0.8008
$(w_1, w_2) = (10, 1.0)$	**0.9753**	0.8005	**0.9746**	**0.8013**
Uncertainty weighting	**0.9753**	**0.8015**	**0.9747**	**0.8015**

Summary

- The application of DL in HEP is growing, but almost of them are for the single task
- The importance of the overall optimization combining such single task might increase in the future.

- Defects in a training of multitask DL models and the mitigation are presented
  - Direct connection of two models causes information bottleneck.

    → Addition of new NN increases information capability and enables to use label for upstream task
  - It is required to tune loss coefficients in the simultaneously training of multiple DL models

    → Application of the methods in multitask learning can tune them automatically.

Backup

Adaptive optimization of loss coefficient : Result

Upstream task: **Cross entropy**, Downstream task: Cross entropy

		Task 1 AUC	Task 2 AUC
No simultaneous training	Step-by-step	**0.9753 ± 0.0002**	0.7969 ± 0.0011
Fixed	$(w_1, w_2) = (0., 1.0)$	0.4717 ± 0.2335	0.7975 ± 0.0009
	$(w_1, w_2) = (0.1, 1.0)$	0.9689 ± 0.0008	0.7993 ± 0.0008
	$(w_1, w_2) = (1.0, 1.0)$	0.9748 ± 0.0003	**0.8013 ± 0.0008**
	$(w_1, w_2) = (10., 1.0)$	**0.9753 ± 0.0003**	0.8005 ± 0.0011
Adaptive	Uncertainty Weighting	**0.9753 ± 0.0003**	**0.8015 ± 0.0011**

Upstream task: **MSE**, Downstream task: Cross entropy

		Task 1 AUC	Task 2 AUC
No simultaneous training	Step-by-step	**0.9746 ± 0.0002**	0.7974 ± 0.0010
Fixed	$(w_1, w_2) = (0., 1.0)$	0.4514 ± 0.2372	0.7977 ± 0.0009
	$(w_1, w_2) = (0.1, 1.0)$	0.9618 ± 0.0010	0.7985 ± 0.0010
	$(w_1, w_2) = (1.0, 1.0)$	0.9717 ± 0.0009	0.8008 ± 0.0008
	$(w_1, w_2) = (10., 1.0)$	**0.9746 ± 0.0002**	**0.8013 ± 0.0010**
Adaptive	Uncertainty Weighting	**0.9747 ± 0.0003**	**0.8015 ± 0.0010**

Step-by-step training cannot propagate sufficient information for downstream task.

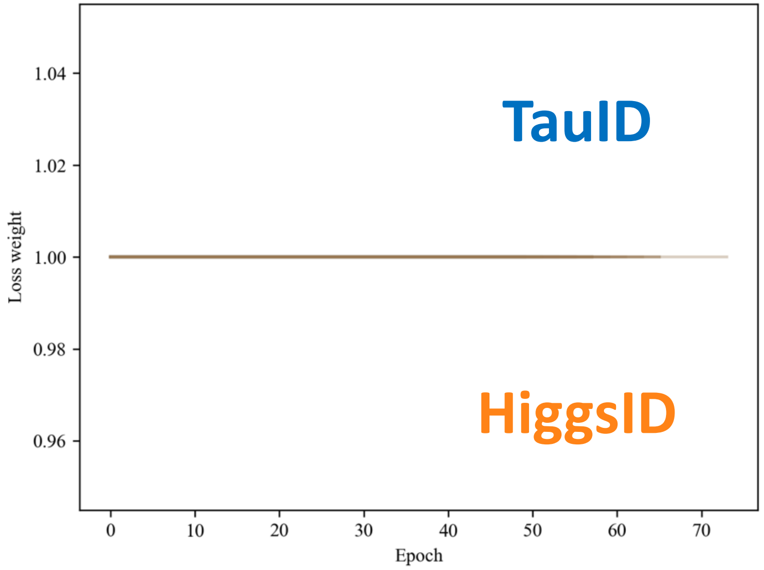
Training w/o upstream loss cannot propagate sufficient information for downstream task.

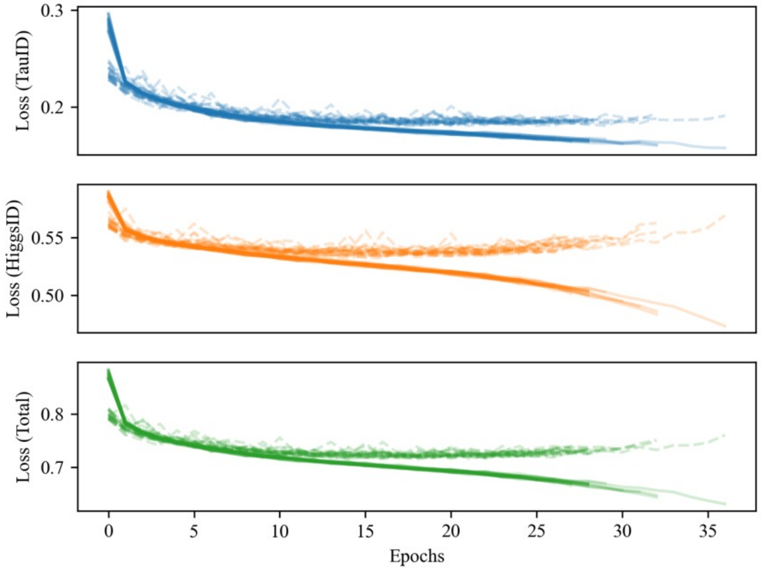
Fixed-coefficient method needs to tune the values.

Uncertainty Weighting has good performance without parameter tuning and independent of loss form.

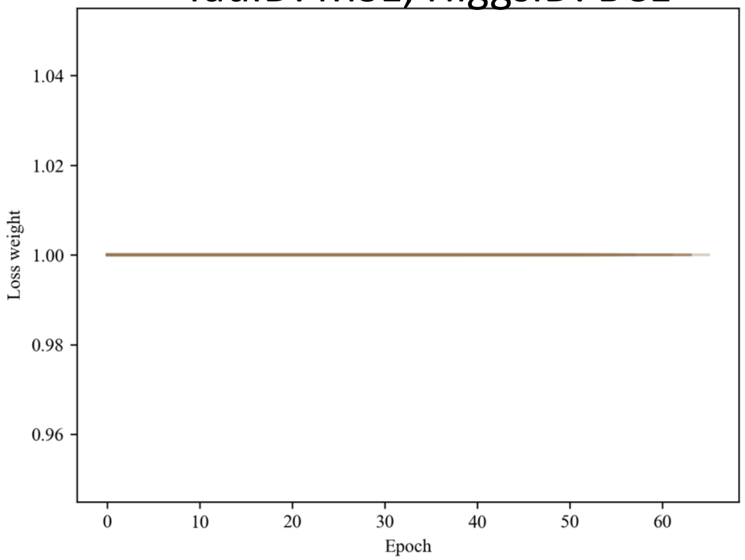
Loss weights (Fixed coefficients)

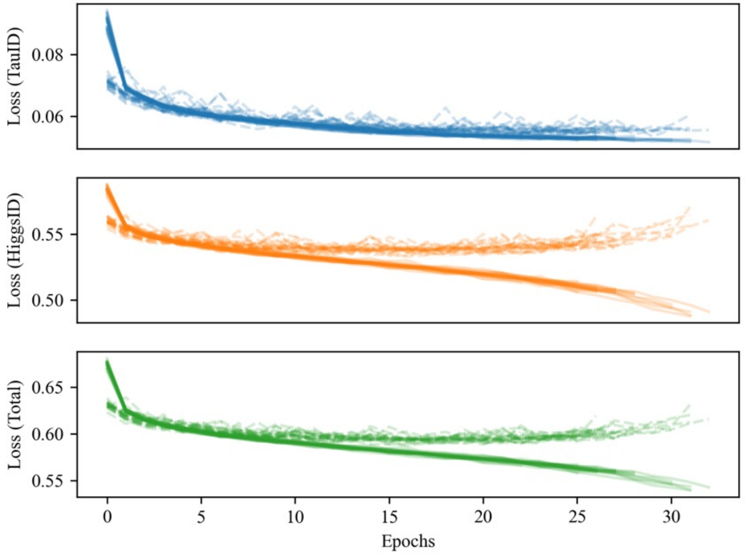
TauID: BCE, HiggsID: BCE





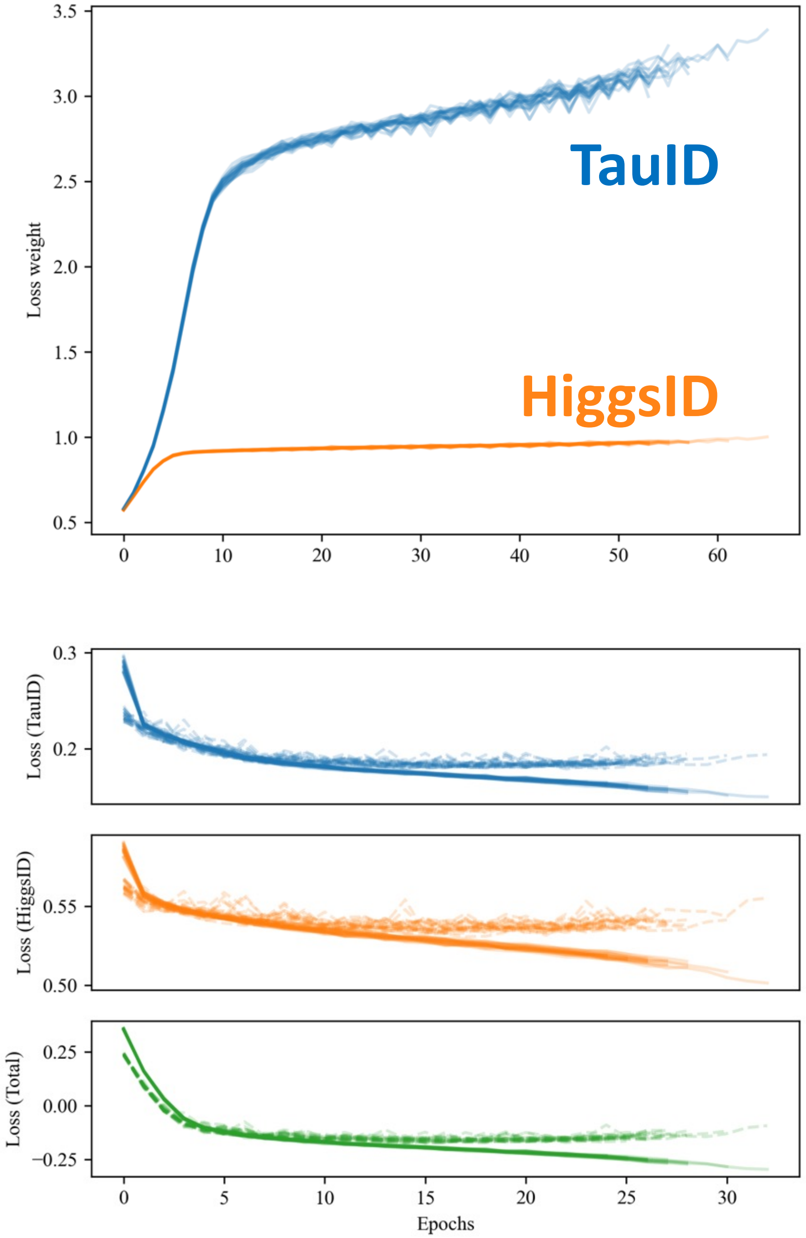
TauID: MSE, HiggsID: BCE



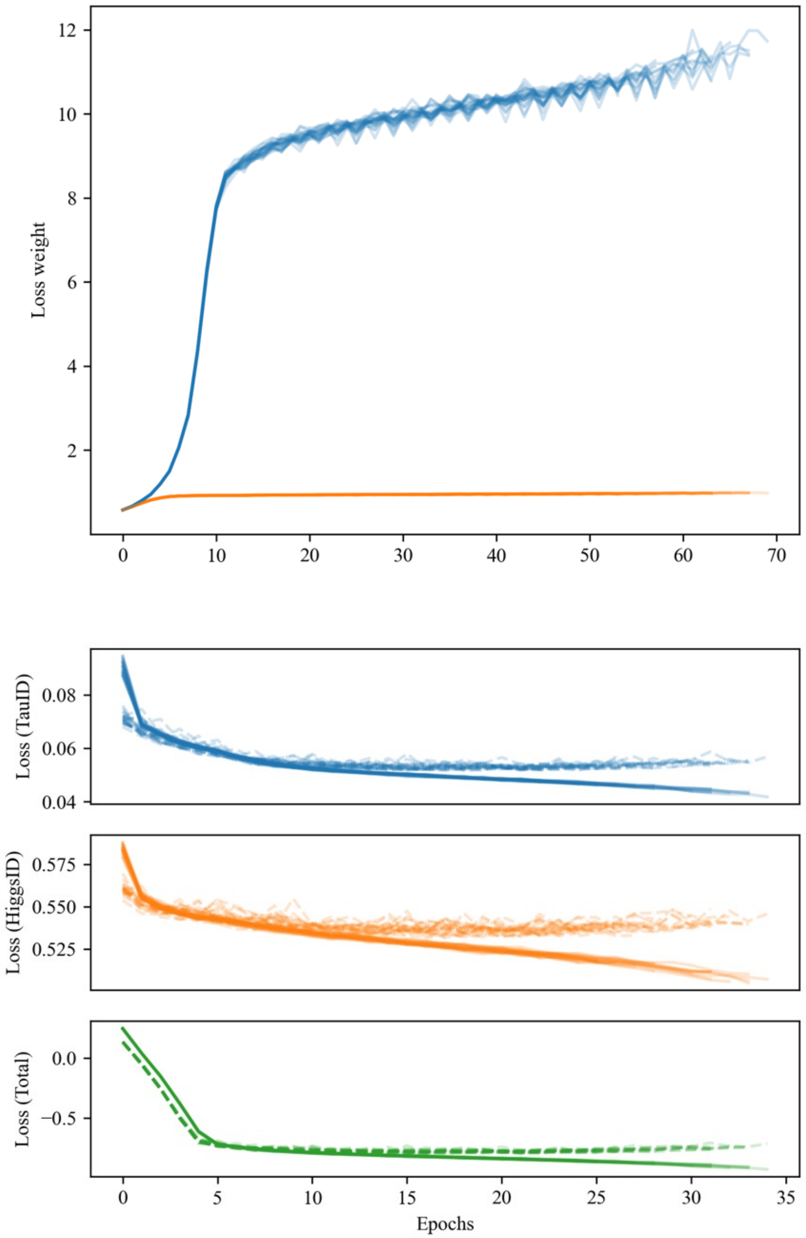


Loss weights (uncertainty weighting)

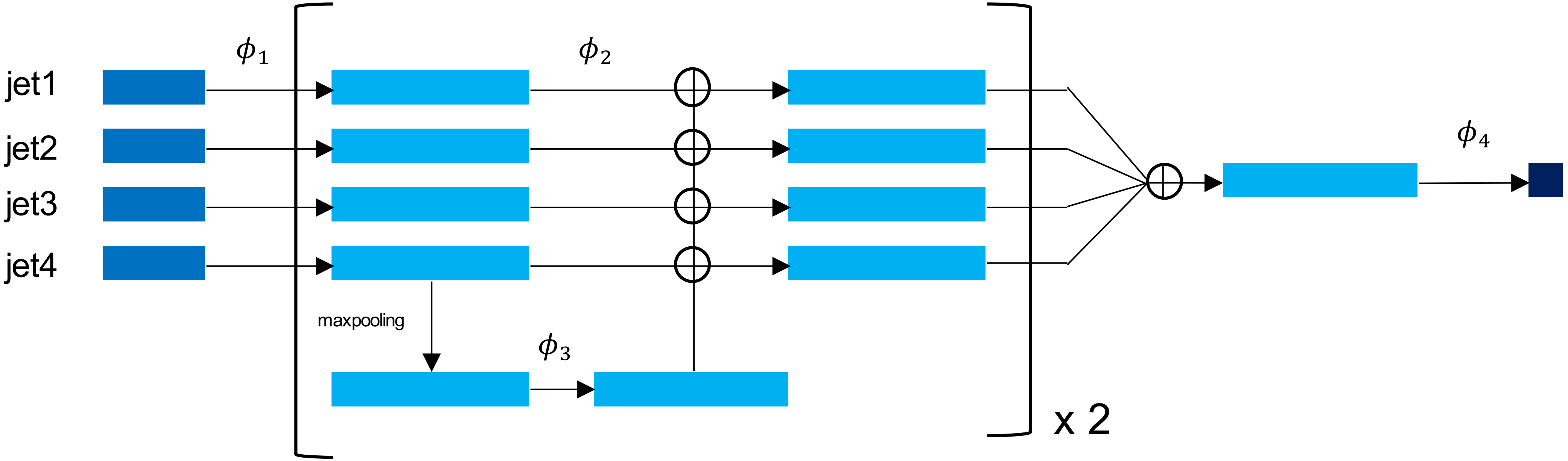
TauID: BCE, HiggsID: BCE



TauID: MSE, HiggsID: BCE



Event Classification Task Model: DeepSets



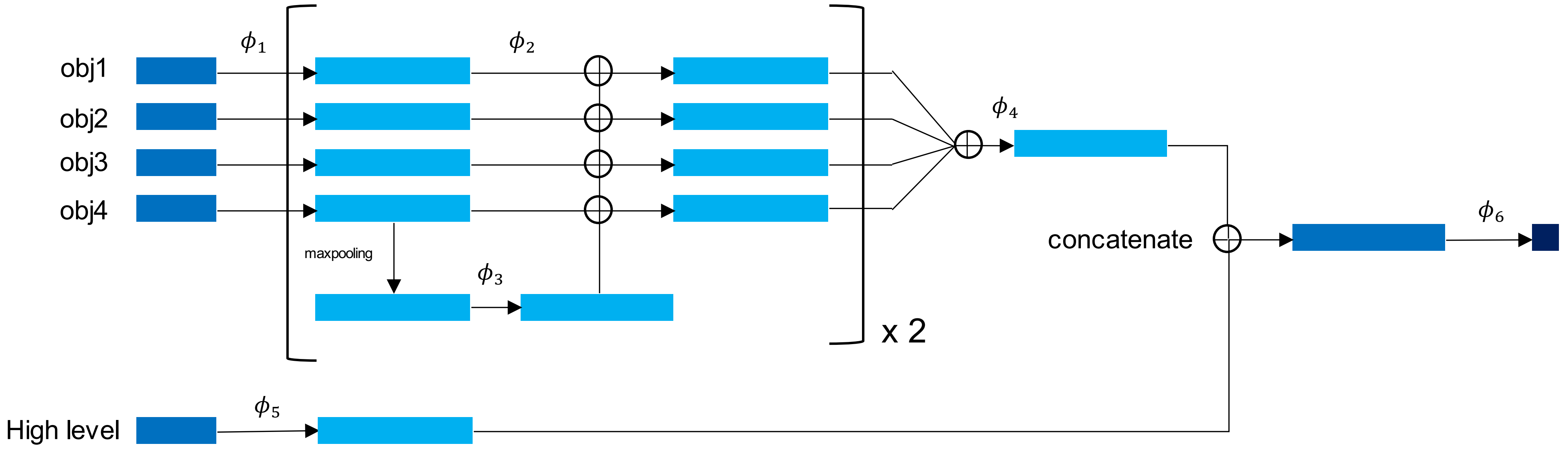
$$\phi_1 = (32, 32, 32), \mathrm{ReLU}$$

$$\phi_2 = (32), \mathrm{Linear}$$

$$\phi_3 = (32), \mathrm{Linear}$$

$$\phi_4 = (64, 32, 1), \mathrm{ReLU}$$

Tau Identification Task Model: DeepSets



$$\phi_1 = (32, 32, 32), \text{ReLU}$$

$$\phi_2 = (32), \text{Linear}$$

$$\phi_3 = (32), \text{Linear}$$

$$\phi_4 = (24), \text{Linear}$$

$$\phi_5 = (128, 128, 16), \text{ReLU}$$

$$\phi_6 = (64, 32, 1), \text{ReLU}$$

Shared latent space when the shared feature's size is 2