國立臺灣大學理學院物理研究所

碩士論文

Department of Physics
College of Science
National Taiwan University
Master's Thesis

利用遷移和元學習提升弱監督搜索效能

Improving the performance of weak supervision searches using transfer learning and meta-learning

陳宗恩

Zong-En Chen

指導教授：蔣正偉 博士

Advisor: Cheng-Wei Chiang Ph.D.

中華民國 113 年 07 月

July 2024

doi:10.6342/NTU202401831

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

利用遷移和元學習提升弱監督搜索效能

Improving the performance of weak supervision searches using transfer learning and meta-learning

本論文係陳宗恩(R10222045)在國立臺灣大學物理所完成之碩士學位論文，於民國 113 年 6 月 26 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Physics on 26 June 2024 have examined a Master's thesis entitled above presented by Zong-En Chen (R10222045) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

（指導教授 Advisor）		
	裴思達 Stathes Paganis	

致謝

首先要先感謝蔣正偉老師的在這四年來的指導與協助，從老師身上學到了不僅是嚴謹的思考邏輯，還有深入的科學觀點。有幸在老師的指導下，我才有機會能見識到粒子物理的美麗與有趣。感謝定國學長在我前期的碩士時期貢獻了許多的協助，從學長身上學到樂於協助及分享的精神。最後也願意在口試的時候提供技術與論文上深入的協助與見解。感謝尚甫、承澤、孫振、俊宏、雋爲、豐仰平時的討論及協助。你們無私的協助與平時有趣的閒談，讓我的碩士生活順利許多且增添許多光彩與樂趣。感謝 Hugues，我很慶幸這碩士時期有機會與你合作研究。從你身上獲得了太多太多的幫助以及指導。我學到最多的是你面對問題時嚴謹的態度，以及不要自滿於任何現狀。這些精神與態度深深地讓我了解做科學研究是多麼的不容易，也因此更需抱持著虛懷若谷的態度面對研究。沒有你的協助，這份研究就不可能完成。感謝 Susan，不管是在大五或者是在碩士班的三年，你的鼓勵總能讓我有動力繼續努力。感謝父母，讓我有機會能夠追尋我的目標。最後感謝我自己，謝謝自己能夠願意投入，並期許未來的自己能夠為現在的我自豪。

doi:10.6342/NTU202401831

中文摘要

弱監督搜尋在原理上具有以下兩個優點：既能夠在實驗數據上進行訓練，又能夠學習到獨特的信號特性。然而，由於在弱監督下成功訓練神經網絡可能需要大量的信號，因此這種搜尋策略的實際應用性受到嚴重限制。在本研究中，我們嘗試開發更高效和更智能的神經網絡，通過利用遷移學習和元學習來從較少的實驗數據信號中學習。其基本思路是首先在模擬數據上訓練神經網絡，學習關鍵概念並成為更高效的學習者。隨後，神經網絡再在實際數據上進行訓練，通過利用從模擬中獲得的知識和概念，期望能夠在學習中需要較少的信號。我們發現，遷移學習和元學習可以顯著提高弱監督搜尋的性能。

doi:10.6342/NTU202401831

Improving the performance of weak supervision searches using transfer learning and meta-learning

Zong-En Chen

Advisor: Cheng-Wei Chiang Ph.D.

Department of Physics

National Taiwan University

Taipei, Taiwan

July 19, 2024

doi:10.6342/NTU202401831

Abstract

Weak supervision searches have in principle the advantages of both being able to train on experimental data and being able to learn distinctive signal properties. However, because successfully training a neural network under weak supervision can require a large amount of signal, the practical applicability of this search strategy is seriously limited. In this study, we try to develop more efficient and smarter neural networks that can learn from less signal in the experimental data by utilizing transfer learning and meta-learning. The general idea is to first train a neural network on simulations, learning critical concepts and becoming a more efficient learner. Subsequently, the neural network is trained on real data and, by exploiting the knowledge and concepts acquired from simulations, should hopefully require less signals to learn. We find that transfer and meta-learning can substantially improve the performance of weak supervision searches.

This study is based on our previous work [1].

doi:10.6342/NTU202401831

Contents

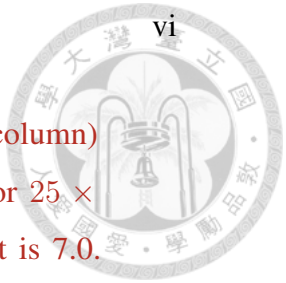
doi:10.6342/NTU202401831

doi:10.6342/NTU202401831

List of Figures

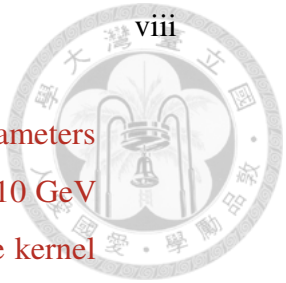
doi:10.6342/NTU202401831

doi:10.6342/NTU202401831



doi:10.6342/NTU202401831

doi:10.6342/NTU202401831



doi:10.6342/NTU202401831

List of Tables

doi:10.6342/NTU202401831

Chapter 1

Introduction

Particle physics investigates elementary particles and their interactions. Central to this field is the Standard Model (SM), a highly successful framework that explains the behavior of fundamental particles and their interactions through the $SU(3)_C \times SU(2)_L \times U(1)_Y$ gauge groups. More precisely, spin-1 gauge particles, including the massless photon γ, the massless gluon g with eight color states, and the massive $W^{\pm}, Z$ bosons, serve as force carriers of electromagnetism, strong interaction, and weak interaction, respectively. Additionally, the three generations of quarks and leptons, comprising both right-handed singlets and left-handed doublets, collectively constitute the matter observed in our universe. Due to the mechanism of spontaneous symmetry breaking of the complex Higgs doublet H, particles acquire their masses by the vacuum expectation value of H, also leaving a spin-0 scalar particle h. The other components of the Higgs doublet ($A_0, H^{\pm}$), known as Goldstone bosons, are absorbed by the $W^{\pm}$ and Z bosons, imparting a non-zero longitudinal mode to these particles. There are 26 input parameters in the SM, and physicists measure and predict these parameters by many experiments [2], especially collider experiments.

Despite the success of the Standard Model, several questions remain unanswered. The SM namely fails to answer:

- What is dark matter?

doi:10.6342/NTU202401831

- How can gravity be incorporated into the model?
- What explains the asymmetry between matter and antimatter?
- How can the Higgs mass hierarchy problem be solved?
- What explains the muon's $g-2$ anomaly?
- What mechanism generates neutrino masses?

These issues highlight the limitations of the SM and underscore the importance of studying its extensions. Experimental research in collider physics and astrophysics has provided substantial insights into these problems.

In recent years, advances in machine learning have provided many opportunities in collider physics. We can utilize neural networks (NNs) to distinguish the signal from the SM background and to search for new particles. In order to create such a neural network, training is necessary. Three main strategies exist, characterized by the way of labeling the data:

1. Fully supervised learning: all data are labeled correctly.
2. Unsupervised learning: none of the data is labeled.
3. Weakly supervised learning: the data are labeled imperfectly.

In fully supervised learning, the training data are correctly labeled as signals and backgrounds. This supervision strategy has been considered and applied in many studies [3, 4, 5, 6]. However, when the goal is to find a new particle that has not been observed yet, the training data must come from simulations instead of experimental data. There are some possible problems with this. First, simulations inherently include imperfections or artifacts. This can lead to the neural network learning from these defects, making the neural network sub-optimal and unpredictable when applied to real data [7]. Second, the reaction of the neural network to a signal that deviates from the expected signal is uncertain. This could

doi:10.6342/NTU202401831

limit the sensitivity of the search to a narrow range of models, possibly causing it to overlook a detectable signal.

Another training strategy is unsupervised learning, where the training data lacks labels. A common approach has been to use autoencoders trained on predominantly background events, using the reconstruction error as a test statistic [8]. More precisely, an autoencoder is a type of neural networks that learns to encode input data into a compressed representation and then decode it back to its original form, trying to minimize the reconstruction error. Although this neural network can directly learn properties of experimental data, there are two notable drawbacks to this method. First, the reconstruction error can sometimes be a weak discriminator [9, 10]. Second, since autoencoders are trained exclusively on background data, they will will not be trained to look for unique characteristics of signal events, hence reducing their discriminative ability.

Weakly supervised learning, using training data with imperfect labels, presents a promising strategy that tries to address the challenges of both the fully supervised and unsupervised learning approaches. In the Classification Without Label (CWoLa) method [7], two sets of experimental data are considered, each assumed to have different mixtures of signal and background events. Under the assumption that the properties of signals (backgrounds) in both mixed datasets are identical, Ref. [7] demonstrated that the most powerful test statistic for distinguishing between these datasets is also the most powerful test statistic for distinguishing pure signals from pure backgrounds. Therefore, a neural network trained to distinguish these datasets naturally becomes proficient at identifying signal events within the data. Such neural networks can train exclusively with the specific signal present in the data and do not need to worry about a difference between the training data and the actual signal, which can be a problem in fully supervised learning. Hence, the CWoLa method can combine both benefits of unsupervised learning (data-driven training) and fully supervised learning (exploiting signal properties). Notably, weak supervision has been implemented in an experimental search by ATLAS (see

doi:10.6342/NTU202401831

Ref. [11]).

Although weakly supervised learning combines the advantages of both fully supervised and unsupervised learning, it faces some practical limitations when the number of signals is limited [12, 13, 14]. In this case, the neural network cannot successfully distinguish signals from backgrounds such that the neural network will indiscriminately cut both signals and backgrounds. At this point, we describe the amount of signal as being below the learning threshold. Worse still, the threshold might be greater than what would be necessary for discovery without using neural networks, and the practical value of such a model is thereby limited. Especially, this situation can happen when the dimensionality of the input is too large, as noted in Ref. [15]. To address this issue, Ref. [16] implemented a solution by providing a simple but effective input to the network. Recent efforts to tackle this challenge can be found in Refs. [17, 18].

In the hope of addressing the limitations of the CWoLa method, the goal of this work will be to create neural networks that can learn from less data. The general idea is for the neural network to use simulations to learn useful concepts and become a better learner, such that it can learn faster once trained on actual data. Despite the limited availability of signal data, it can be easy to generate simulations of it. The neural network can first learn from the simulations and then understand critical concepts. Subsequently, the neural network can be trained on real data more efficiently via knowledge obtained from simulations. Hence, the neural network could learn faster and require less real data in the training. In this study, we will consider and use transfer learning and meta-learning to address this issue.

The basic idea of transfer learning involves the transfer of knowledge or expertise gained from solving a previous task to improve the learning or performance of a neural network model on a different but related task. This technique is particularly useful when there is limited labeled data available for the target task. In transfer learning, the neural network model is pretrained on a large and general

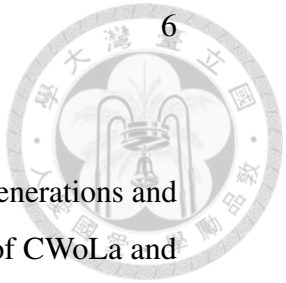
doi:10.6342/NTU202401831

dataset, called source data, and is adapted or fine-tuned to a small and specific dataset, called target data. Usually, the source dataset and the target dataset have many similarities. Hence, the neural network can first learn the fundamental concepts in the source dataset, and then reuse the concepts learned previously to learn more efficiently. This training strategy has been applied in many studies (see Refs. [13, 14, 19, 20, 21, 22, 23, 24, 25, 26]). In this study, we will utilize pretraining and finetuning strategies for transfer learning, and the details of the strategy will be explained in Chap. 4.

The basic idea of meta-learning involves enhancing neural networks by leveraging knowledge gained from training with multiple tasks (see Ref. [27] for an example application in high energy physics and Ref. [28] for a review). Unlike traditional machine-learning approaches that focus on learning from specific datasets and then adapting to the target task, meta-learning focuses on equipping models with the ability to learn how to learn. In meta-learning, the emphasis lies in acquiring higher-level understanding, often referred to as "meta-knowledge", about the learning process itself. This meta-knowledge includes insights into various aspects of learning, such as the characteristics of different tasks, the relationships between tasks, and the most effective strategies for adapting to new tasks. The basic approach is to let the neural network learn from multiple tasks. Ideally, the neural network can gain the meta-knowledge with these tasks, and then become a more efficient learner and require less data for the target task. In this work, we use meta-transfer learning as our meta-learning strategy, and the details will be explained in Chap. 5.

In this study, we find the following results. First, transfer learning successfully enhances the performance of the neural networks under weak supervision. The learning thresholds can be reduced significantly and the amount of signal necessary for discovery can sometimes be several times smaller. Second, meta-transfer learning can further improve the performance of the neural networks. However, the improvements between transfer learning and meta-transfer learning are smaller

doi:10.6342/NTU202401831



than the improvements between CWoLa and transfer learning.

This study is organized as follows. Chap. 2 presents the events generations and the image prepossessing for jet images. Chap. 3 explains the idea of CWoLa and the difficulties of CWoLa for learning thresholds. Chap. 4 and Chap. 5 presents the details of the strategies and results for transfer learning and meta-learning. Finally, Chap. 6 summarizes this study.

doi:10.6342/NTU202401831

Chapter 2

Events generation

2.1 Signal and background generation

In this study, we use the Hidden Valley (HV) model [29, 30] as our benchmark (see Ref. [31] for a review). It consists of a set of particles charged under a new confining group and that somehow communicate with the SM sector. If produced at colliders and relatively light, these particles will shower and create collimated sprays of dark hadrons. Some of these will in turn decay to SM particles and create an object that can potentially mimic a QCD jet. These are known as dark showers. Dark hadrons can provide many potential dark matter candidates [32, 33, 34] and have been the focus of multiple experimental searches [35, 36, 37, 38].

The `Pythia` HV module is used for simulating dark showers, providing a broad range of signals due to its numerous adjustable parameters. This flexibility makes the module particularly advantageous for transfer learning and meta-learning. Specifically, the signal process considered is $pp \to Z' \to \bar{q}_D q_D$. The dark quarks q_D are a set of fermions charged under a new confining gauge group $SU(3)_{dark}$ but neutral under the SM gauge groups $SU(3)_C \times SU(2)_L \times U(1)_Y$. These dark quarks are assumed to be degenerate in mass for simplicity. The Z' particle is a massive Abelian gauge boson that interacts with both SM quarks and dark quarks. Hence, the signature of the final state is a pair of dark jets with an invariant mass

doi:10.6342/NTU202401831

M_{jj} consistent with the mass of Z' boson.

Once produced, the dark quarks are showered and hadronized by `Pythia` 8.307. After dark showering, the resulting dark hadrons are either dark vector mesons ρ_D or dark pseudo-scalar mesons π_D. We follow the recommendations from Ref. [31], and the ratio of their masses is set:

$$\frac{m_{\pi_D}}{\Lambda_D} = 5.5\sqrt{\frac{m_{q_D}}{\Lambda_D}}, \quad \frac{m_{\rho_D}}{\Lambda_D} = \sqrt{5.76 + 1.5\frac{m_{\pi_D}^2}{\Lambda_D^2}}, \quad m_{q_{\text{const}}} = m_{q_D} + \Lambda_D, \tag{2.1}$$

where m_{q_D} and $m_{q_{\text{const}}}$ are the current and constituent mass of the dark quarks respectively and Λ_D is the dark confining scale for $SU(3)_{dark}$. Note that the dark quark mass in the HV settings of `Pythia` is the constituent mass. In order to verify the validity of Eq. (2.1), by assuming the confining scale and quark mass to be 300 MeV and 3 MeV respectively (which are the values in the SM for the QCD confining scale and the average of masses of up and down quarks), the masses of pions and rhos are 160 MeV and 750 MeV respectively. These are close to the values of the SM. Furthermore, when the relation $m_{\rho_D} \geq 2m_{\pi_D}$ holds, the decay of $\rho_D \to \pi_D\pi_D$ is allowed. From Eq. (2.1), we have

$$m_{\rho_D} = \sqrt{5.76\Lambda_D^2 + 1.5m_{\pi_D}^2} \geq 2m_{\pi_D}. \tag{2.2}$$

Hence, the decay of $\rho_D \to \pi_D\pi_D$ is allowed if $m_{\pi_D}/\Lambda_D < 1.52$, and we consider the two scenarios from the different decay channels of ρ_D as our benchmarks.

In the first scenario, where $m_{\rho_D} \geq 2m_{\pi_D}$ and the decay $\rho_D \to \pi_D\pi_D$ is permitted, this decay mode dominates with a branching ratio of effectively 1. We define seven benchmarks within this scenario, each distinguished by different Λ_D values and fixing a constant mass ratio of $m_{\pi_D}/\Lambda_D = 1$. The selected values of Λ_D are 1, 5, 10, 20, 30, 40, and 50 GeV, and the corresponding masses of π_D, ρ_D, q_D, and q_{const} determined by Eq. (2.1). For simplicity, we assume exclusive decay of the dark pions to SM $d\bar{d}$ pairs. This scenario is denoted as Indirect Decay (ID).

In the second scenario, where $m_{\rho_D} < 2m_{\pi_D}$ and the decay $\rho_D \to \pi_D\pi_D$ is forbidden, we also define seven benchmarks, each distinguished by different Λ_D values and fixing a constant mass ratio of $m_{\pi_D}/\Lambda_D = 1.8$. The selected values

doi:10.6342/NTU202401831

of Λ_D are the same as those of the first scenario: 1, 5, 10, 20, 30, 40, and 50 GeV. Again, the corresponding masses of π_D, ρ_D, q_D, and q_{const} are determined by Eq. (2.1). We assume, for simplicity, exclusive decay of both the dark pions and dark rho mesons to SM $d\bar{d}$ pairs in this scenario, referred to as Direct Decay (DD). Table 2.1 lists the parameter values for both scenarios.

The additional relevant signal parameters in `Pythia` are specified as follows: The mass of Z' is set to 5.5 TeV, resulting in an invariant mass of the leading two jets of approximately 5.2 TeV. The slight discrepancy is attributed to part of the constituents falling outside the reconstructed jets. Fig. 2.1 illustrates the distribution of the invariant mass of the two leading jets, denoted as M_{jj}. The decay width of Z' is set to 10 GeV, ensuring that there is no significant peak broadening that could negatively impact the search. The settings of the remaining `Pythia` parameters are detailed in Table 2.3.

Next, the dominant background is expected to be from the pair production of QCD jets, denoted as $pp \to jj$. These background events are generated at parton level by `Madgraph` 2.7.3 [39] and subsequently hadronized by `Pythia` 8.307. For simplicity, only leading order jet pair production is considered. The initial cuts listed in Table 2.2 are used in `Madgraph` to enhance the generating efficiency. It has been verified that these preliminary cuts are weak enough to avoid any significant impact on the relevant parts of the distribution. The parton distribution function adopted for both signal and background event generations is NN23LO1 [40]. Default settings within `Pythia` are used for the hadronization of background events.

Both signal and background events undergo detector simulation by using `Delphes` 3.4.2 [41], and jet reconstruction is dealt with via the anti-k_T clustering algorithm implemented in `FastJet` 3.3.2 [42], with a jet radii of $R = 0.8$. This choice is different from the default value of 0.5 to accommodate the larger jet radius characteristic of signal jets originating from dark showers, ensuring that at least 90% of the constituents are included within the jet. After detector simulations,

doi:10.6342/NTU202401831

Scenarios	Λ_D	m_{π_D}	m_{ρ_D}	$m_{q_{\text{const}}}$
ID	1	1	2.69	1.03
ID	5	5	13.47	5.17
ID	10	10	26.94	10.33
ID	20	20	53.89	20.66
ID	30	30	80.83	30.99
ID	40	40	107.78	41.32
ID	50	50	134.72	51.65
Scenarios	Λ_D	m_{π_D}	m_{ρ_D}	$m_{q_{\text{const}}}$
DD	1	1.8	3.26	1.11
DD	5	9	16.29	5.54
DD	10	18	32.59	11.07
DD	20	36	65.18	22.14
DD	30	54	97.77	33.21
DD	40	72	130.35	44.28
DD	50	90	162.94	55.36

Table 2.1: Parameters for the different benchmarks in the indirect decaying (ID) and direct decaying (DD) scenarios. All values are in GeV units.

the selection criteria listed in Table 2.2 are applied. Notably, the Signal Region (SR) and Sidebands (SB) are defined and used in the later CWoLa procedure. Via fixing the integrated luminosity, the background in the SR and in the SB contain 20k and roughly 21k events passing the SR and SB selection cuts respectively. The corresponding integrated luminosity is 147.3 fb^{-1}, which is close to the integrated luminosity used in Run 2 of the LHC. The cross-section, the cut efficiency, and the number of events for the background are listed in Table 2.4.

To verify the requirement of CWoLa that the properties of signals (backgrounds) are identical in both mixed datasets, we consider using the high-level and low-level physical quantities to examine the requirement. The generalized angularities

doi:10.6342/NTU202401831

Preliminary cuts in Madgraph for the SM
$\sqrt{s} = 13$ TeV
Both P_T of the leading two jets > 700 GeV
Both η of leading two jets $\|\eta_j\| < 2.2$
$M_{jj} > 3000$GeV
Selection criteria after Delphes
Number of jets $n_j \geq 2$
Both P_T of the leading two jets > 750 GeV
Both η of leading two jets $\|\eta_j\| < 2$
SR = $\{M_{jj} \in [4700, 5500]\}$
SB = $\{M_{jj} \in [4400, 4700] \cup [5500, 5800]\}$

Table 2.2: Parameters in Madgraph and the selection criteria after Delphes.

HV parameters in Pythia	
HiddenValley: alphaOrder	1
HiddenValley: nFlav	3
HiddenValley: Ngauge	3
HiddenValley: pTminFSR	$1.1\Lambda_D$
HiddenValley: separateFlav	on
HiddenValley: aLund	0.1
HiddenValley: bmqv2	1.9
HiddenValley: rFactqv	1.0
HiddenValley: probVector	0.75
HiddenValley: fragment	on
HiddenValley: FSR	on

Table 2.3: Parameters for dark showering in Pythia.

(GAs) and the jet images are considered for the high-level and low-level quantities respectively, and jet images will be discussed in Sec. 2.2 and be our input data in

doi:10.6342/NTU202401831

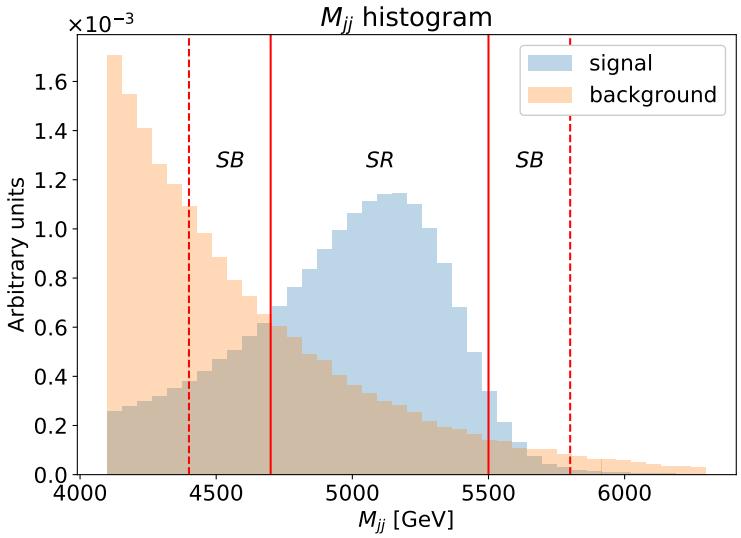


Figure 2.1: Dijet invariant mass distributions for the indirect decaying scenario with $\Lambda_D = 10$ GeV and for the SM background. Distributions are normalized to unity. Both signal and background satisfy the selection criteria of Table 2.2 except for the SR or SB conditions.

	Efficiency of the cut	σ after the cut	number of events
Background in SR	0.020	135.8 fb	20000
Background in SB	0.021	142.3 fb	20957

Table 2.4: The cross-section and the efficiency of selection cut listed in Table 2.2 for the SM background. We set the integrated luminosity 147.3 fb^{-1}. The cross-section of the background in the `Madgraph` level is 6.8 pb.

neural networks. The generalized angularities are commonly used to discriminate the gluon and quark jets efficiently (see Refs. [43, 44, 45, 46]). The GAs are denoted as λ_β^κ with different choice of κ and β and calculated by:

$$\lambda_\beta^\kappa = \sum_{i \in jet} z_i^\kappa \theta_i^\beta, \tag{2.3}$$

$$z_i = \frac{P_{T,i}}{\sum_{i \in jet} P_{T,i}}, \tag{2.4}$$

doi:10.6342/NTU202401831

$$\theta_i = \frac{\Delta R_i}{R}, \tag{2.5}$$

$$\Delta R_i = \sqrt{(\phi_i - \phi_{jet})^2 + (\eta_i - \eta_{jet})^2}, \tag{2.6}$$

where the $P_{T,i}$ is the transfer momentum of the jet constituent, ΔR_i is the pseudo-rapidity/azimuth distance of the jet constituent to the jet-axis, R is the jet reconstruction radius which is set to be 0.8, and ϕ_{jet} and η_{jet} are the PT-weighted pseudo-rapidity and azimuth angles defined in Eq. (2.8). The choices of (κ, β) are set to be $(0, 0)$, $(2, 0)$, $(1, 0.5)$, $(1, 1)$, $(1, 2)$, respectively called multiplicity, $(p_T^D)^2$ [47], Les Houches Angularity (LHA) [48, 49], width, and mass.[1] Fig. 2.2 shows that the distributions of signal (background) in SR and SB are highly similar, satisfying the requirement of CWoLa mentioned in Sec. 3.1. Fig. 2.3 shows that the correlation coefficients of GAs between the two leading jets are quite small.

2.2 Image preprocessing

In order to illustrate the power of transfer and meta-learning, we will use jet images as input to the neural networks. Such high dimensional inputs can be challenging for weak supervision, but we will show that our procedure still works under these conditions. The ability to adjust the resolution and therefore the input size will also prove useful to illustrate certain features.

To use images as input data, the two leading jets in P_T are converted into jet images according to the following procedure [50, 51, 52]: translation, rotation, flipping, and pixelization.

1. Translation: each jet constituent is translated so that the center of the jet image is along the jet axis. That is

$$(\eta_i, \phi_i) \to (\eta_i', \phi_i') = (\eta_i - \eta_{jet}, \phi_i - \phi_{jet}), \tag{2.7}$$

[1] Strictly speaking, it is not the regular mass. It is the mass-squared over energy-squared in the soft-collinear limit. We just adopt the same names listed in Ref. [7].

doi:10.6342/NTU202401831

$$\eta_{jet} = \frac{\sum_k \eta_k P_{T,k}}{\sum_k P_{T,k}}, \quad \phi_{jet} = \frac{\sum_k \phi_k P_{T,k}}{\sum_k P_{T,k}}, \tag{2.8}$$

where η_{jet} and ϕ_{jet} are weighted by transverse momenta. After the translation, the PT-weighted η_{jet} and ϕ_{jet} are zero.

2. Rotation: define the PT-weighted mass matrix M,

$$M = \frac{\sum_i \begin{pmatrix} P_{T,i}\eta_i \\ P_{T,i}\phi_i \end{pmatrix} \begin{pmatrix} P_{T,i}\eta_i & P_{T,i}\phi_i \end{pmatrix}}{\sum_i P_{T,i}^2} = \frac{1}{\sum_i P_{T,i}^2} \begin{pmatrix} \sum_i P_{T,i}^2\eta_i^2 & \sum_i P_{T,i}^2\eta_i\phi_i \\ \sum_i P_{T,i}^2\eta_i\phi_i & \sum_i P_{T,i}^2\phi_i^2 \end{pmatrix}. \tag{2.9}$$

Because the M matrix is symmetric, there is an orthogonal matrix U which can diagonalize the matrix M,

$$M_{diag} = UMU^{-1} = \begin{pmatrix} M_{11} & 0 \\ 0 & M_{22} \end{pmatrix}, \tag{2.10}$$

where M_{11} and M_{22} are the principle values (eigenvalues) of the matrix M. We choose $M_{11} \geq M_{22}$ by convention, such that the leading principle axis is along the η direction after rotation. Then the new η_i', ϕ_i' are defined by

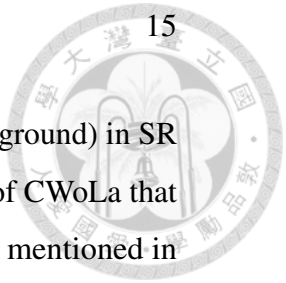
$$\begin{pmatrix} \eta_i' \\ \phi_i' \end{pmatrix} = U \begin{pmatrix} \eta_i \\ \phi_i \end{pmatrix}. \tag{2.11}$$

3. Flipping: the image is flipped such that the highest P_T constituent is in the first quadrant (upper-right plane).

4. Pixelization: The jet constituents are pixelated with resolutions of either 25×25, 50×50 or 75×75. The range of η and ϕ are both from -1 to 1.

Fig. 2.4 and Fig. 2.5 present the jets before and after preprocessing and the corresponding average histograms. The figures also highlight the jet radius $R = 0.8$ used in the jet reconstruction process. The capability to adjust the resolution will be beneficial for illustrating specific features and influencing the learning thresholds.

doi:10.6342/NTU202401831



Last, the average plots show that the average images of signal (background) in SR and SB are highly similar to each other, satisfying the requirement of CWoLa that the distribution of signal (background) in SR and SB are identical, mentioned in Sec. 3.1.

doi:10.6342/NTU202401831

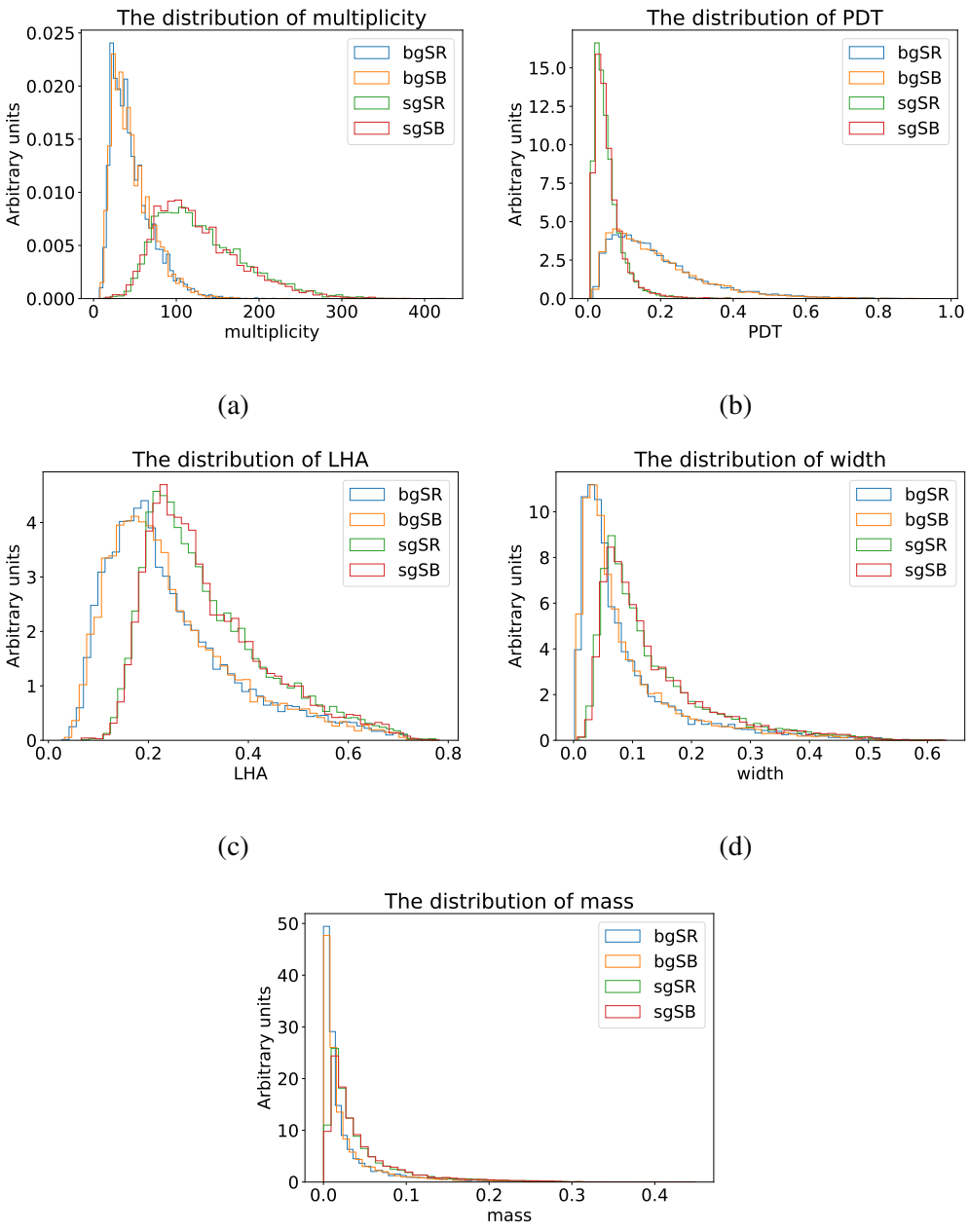


Figure 2.2: The distributions of five generalized angularities for the first leading jet. The labels of bgSR, bgSB, sgSR, and sgSB are the 10k events of signals and backgrounds in the SR and SB. The events are after the cut listed in Table 2.2 and distributions are normalized to be unity. The benchmark of signals is the indirect decaying scenario with $\Lambda_D = 10$ GeV.

doi:10.6342/NTU202401831

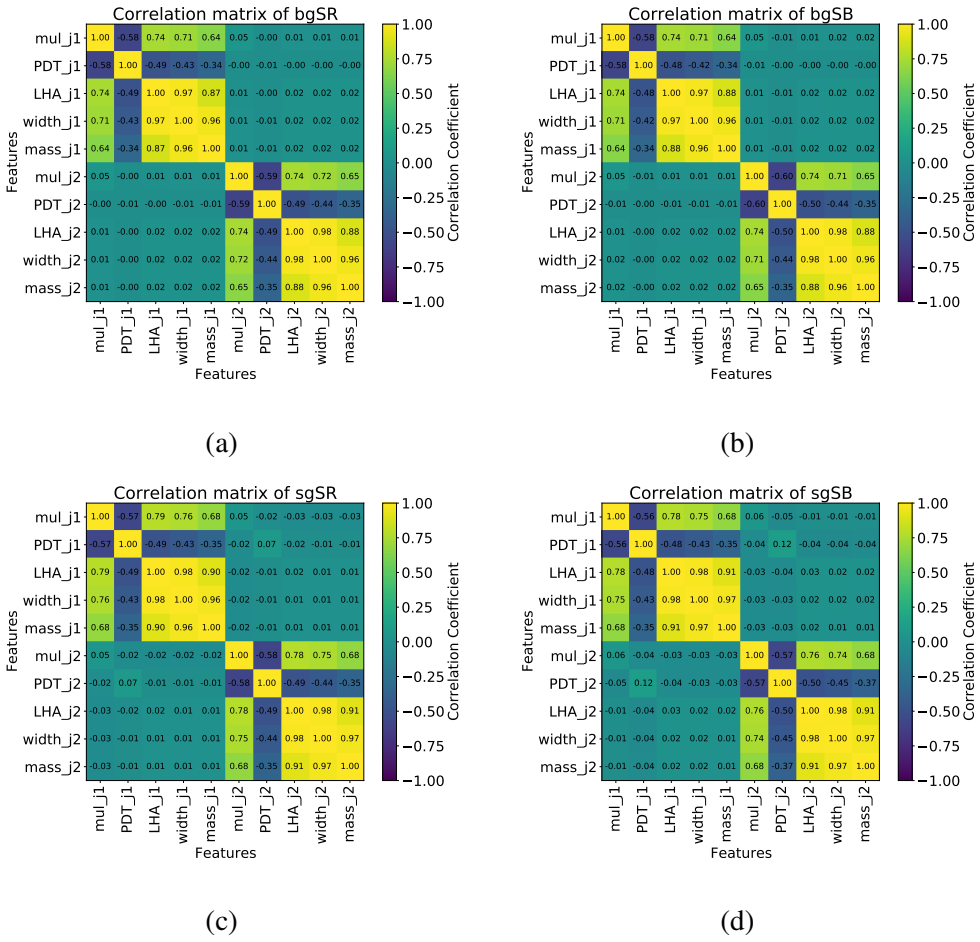
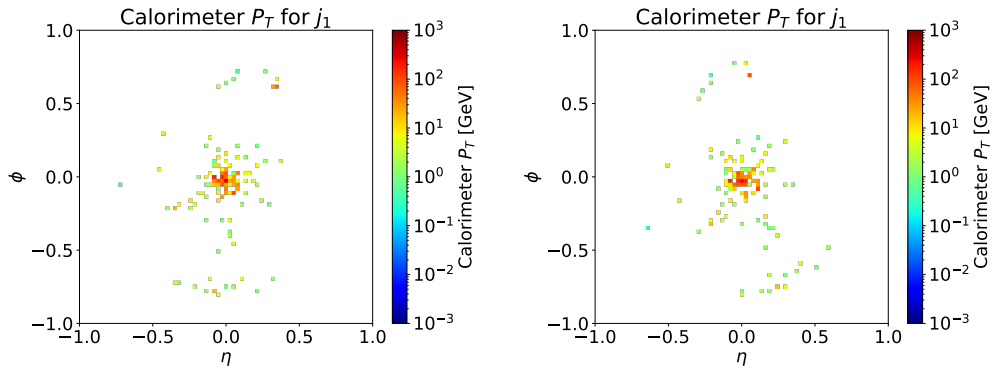


Figure 2.3: The correlation coefficient matrix of bgSR, bgSB, sgSR, and sgSB for the five GAs of the leading two jets. The events are after the cut listed in Table 2.2. The benchmark of signals is the indirect decaying scenario with $\Lambda_D = 10$ GeV. Note that the upper-right and lower-left regions in each subplot are the correlation coefficients between the leading two jets.

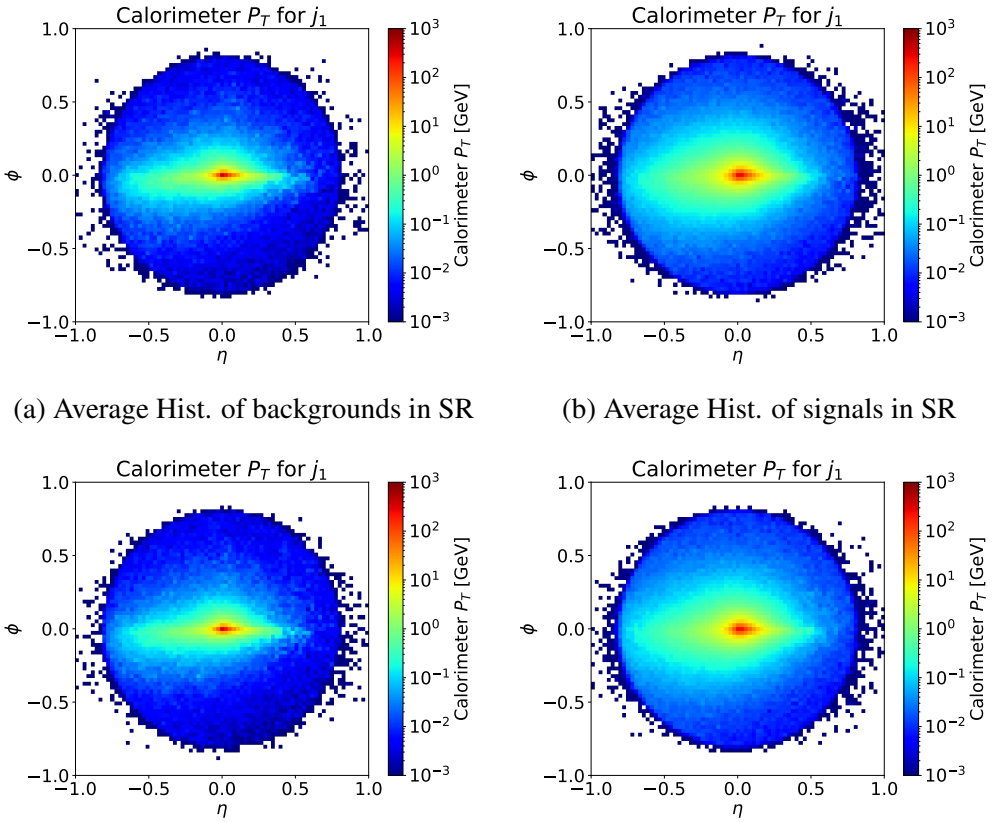
doi:10.6342/NTU202401831



(a) Before preprocessing (b) After preprocessing

Figure 2.4: (a) A 2D P_T histogram for one signal event in the SR before rotation and flipping. (b) A 2D P_T histogram of the same event after complete preprocessing. These plots are for the leading jet with 75×75 resolution and the ID scenario with $\Lambda_D = 10$ GeV.

doi:10.6342/NTU202401831



(a) Average Hist. of backgrounds in SR

(b) Average Hist. of signals in SR

(c) Average Hist. of backgrounds in SB

(d) Average Hist. of signals in SB

Figure 2.5: (a) The average histogram for 10k background events in the SR after preprocessing. (b) The average histogram for 10k signal events in the SR after preprocessing. (c) The average histogram for 10k background events in the SB after preprocessing. (d) The average histogram for 10k signal events in the SB after preprocessing. These plots are for the leading jet with 75×75 resolution and the ID scenario with $\Lambda_D = 10$ GeV.

doi:10.6342/NTU202401831

Chapter 3

Classification without labels (CWoLa)

3.1 Theoretical perspectives

In this section, we briefly review classification without labels (CWoLa) following Ref. [7]. Let's define a classifier $F : \mathbf{x} \to z \in \mathbb{R}$, where $\mathbf{x}$ is a vector of observables used to discriminate signals from backgrounds, and z is a real number. The type of $\mathbf{x}$ can be the high-level physics parameters, like M_{jj}, η_{jet}, and ϕ_{jet}, or the low-level physics parameters, like jet images. The higher (lower) values of z mean the $\mathbf{x}$ of the event is more signal-like (background-like). By the Neyman-Pearson lemma [53], the optimal classifier $F_{optimal}$ is the likelihood ratio: $F_{optimal}(\mathbf{x}) = p_S(\mathbf{x})/p_B(\mathbf{x})$, where the p_S and p_B are the probability density functions of $\mathbf{x}$ for the signal and the background. Hence, optimally training a neural network is to make an NN approach the optimal classifier $F_{optimal}$ as much as possible.

In full supervision, each event carries the correct label $y_i \in \{S, B\}$. Also, the output z from the classifier is adjusted to be from 0 to 1 by convention. A neural network is trained with training data to minimize the loss function to become a better classifier. One of the common choices of loss functions is the binary

doi:10.6342/NTU202401831

cross-entropy function $\mathcal{L}_{BC}$:

$$\mathcal{L}_{BC} = -\frac{1}{N}\sum_{j=1}^{N}\left[\mathbb{I}(y_j = S)\log F(\mathbf{x}_j) + (1 - \mathbb{I}(y_j = S))\log\left(1 - F(\mathbf{x}_j)\right)\right], \quad (3.1)$$

where $\mathbb{I}$ is the indicator function for signals, and N is the size of the batch in the training data. Theoretically, given sufficiently large training samples, a flexible model parameterization, and an appropriate minimization process, the learned neural network could approach $F_{optimal}$.

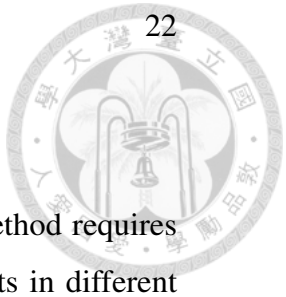
In weak supervision, we assume there are two mixed samples Ω_1 and Ω_2 containing different mixtures of signals and backgrounds. Label the signal fractions as f_1 and f_2 respectively, where we can always assume $f_1 > f_2$. Assume the distributions of signals p_S (backgrounds p_B) are identical within both mixed samples Ω_1 and Ω_2. The likelihood that an event is from Ω_1 is then $p_{\Omega_1} = f_1 p_S + (1 - f_1)p_B$ and similarly for Ω_2. Then, Ω_1 and Ω_2 are more signal-like and background-like respectively. By the Neyman-Pearson lemma, there is an optimal classifier $F_{\Omega_1,\Omega_2} = p_{\Omega_1}/p_{\Omega_2}$ for distinguishing Ω_1 and Ω_2. By the same lemma, there is also another optimal classifier $F_{S,B} = p_S/p_B$ for distinguishing signals and backgrounds. Then,

$$F_{\Omega_1,\Omega_2} = \frac{p_{\Omega_1}}{p_{\Omega_2}} = \frac{f_1 p_S + (1 - f_1)p_B}{f_2 p_S + (1 - f_2)p_B} = \frac{f_1 F_{S,B} + 1 - f_1}{f_2 F_{S,B} + 1 - f_2} \quad (3.2)$$

$$= \frac{f_1}{f_2} + \frac{1 - f_1/f_2}{f_2 F_{S,B} + (1 - f_2)} = \frac{f_1}{f_2} - \frac{1}{f_2}\frac{f_1 - f_2}{f_2 F_{S,B} + (1 - f_2)}. \quad (3.3)$$

The classifier F_{Ω_1,Ω_2} is a monotonically increasing function of $F_{S,B}$, i.e. F_{Ω_1,Ω_2} has the same ability to distinguish signal from background as $F_{S,B}$. Therefore, if the classifier F_{Ω_1,Ω_2} is optimal for distinguishing Ω_1 and Ω_2, then it is also optimal for distinguishing signals and backgrounds. More importantly, the neural network does not require any information about f_1 or f_2, the only requirement is the two mixed datasets of signals and backgrounds with different fractions. In practice, the neural network is trained to distinguish with mixed samples and thereby learns the difference between signals and backgrounds to become the classifier that distinguishes signals well from backgrounds.

doi:10.6342/NTU202401831



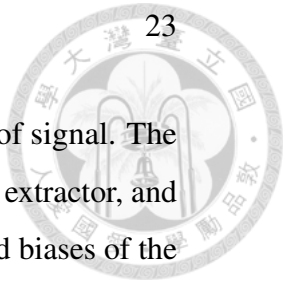
3.2 The implementation of CWoLa

As discussed in the Introduction and Section 3.1, the CWoLa method requires two mixed samples containing both signal and background events in different proportions. In our study, we utilize the resonance peak resulting from the decay of Z' shown in Fig. 2.1. The neural network is trained to discriminate the signal region and sideband regions presented in the figure. This section provides details of our implementation of this procedure, partially inspired by the approach outlined in Ref. [16].

On one hand, the background in the signal region consists of 20k events passing the SR selection cuts listed in Table 2.2. The number of background events in the sidebands is determined using the same integrated luminosity as the signal region. On the other hand, the signal amount in the SR is varied throughout the analysis, resulting in a pre-neural network cut significance ranging from 0 to 7, while the signal in the SBs is adjusted accordingly with the integrated luminosity. Four-fifths of these events are utilized to update neural network parameters, while the remaining one-fifth serves as validation data to monitor validation loss and prevent overfitting. This training data is treated as pseudo-experimental data. During training, the callbacks function is used to save the best model based on validation loss. To test the performance of the CWoLa method, additional 20k signal events and 20k backgrounds, both passing the signal region cut.

For the format of training data, we use jet images of the two leading jets. The distributions of each jet image are independently batch normalized. Each jet image then passes through a common sub-Convolutional Neural Network (subCNN), with each returning a single real-valued number between 0 and 1. The final output of the neural network is the product of these two numbers. The subarchitecture and training procedures are described in Table 3.1, and Fig. 3.1 illustrates the model architecture. All neural networks are implemented using `Keras` [54] with the `TensorFlow` [55] backend. We also explored the possibility of using two distinct networks, but found this alternative typically gave inferior results, as discussed in

doi:10.6342/NTU202401831



Sec. 3.3 and shown in Fig. 3.3. This appears to be due to the lack of signal. The convolutional part of the neural network is referred to as the feature extractor, and its weights and biases are collectively labeled as Θ. The weights and biases of the dense layers are collectively labeled as θ.

In order to evaluate the performance of the NN, we use the significance formula [56]

$$\sigma = \sqrt{2\left((N_s + N_b)\log\left(\frac{N_s}{N_b} + 1\right) - N_s\right)}, \tag{3.4}$$

where N_s and N_b are respectively the numbers of signal and background before and after the NN classification. From the receiver operating characteristic (ROC) curve with testing data after training, we choose specific background efficiencies of $\epsilon_b = 10\%, 1\%, 0.1\%$ and calculate the corresponding signal efficiencies ϵ_s. To examine the robustness of the neural network, the training is performed 10 times for each significance value, including resampling new events in each pseudo-experiment, and averaged. The standard deviations are computed and correspond to fluctuations from both the training and the sampling.

3.3 Discussion

Fig. 3.2 shows two benchmarks with three different resolutions each. Several comments are in order. First, the different curves display a threshold below which the neural network fails to learn from the data. This threshold, discussed in the Introduction, corresponds to the upward turn of the curves around 2 to 4σ. Below this threshold, the NN cuts background and signal indiscriminately, resulting in worse significance than without employing the NN. Second, increasing the resolution tends to move the position of the threshold to higher significance. This is because classifying a higher-resolution image is a more difficult task, requiring more parameters to be learned by the NN.

Fig. 3.3 shows two benchmarks with 25×25 resolutions for different CNN subarchitectures. As mentioned in Sec. 3.2, since the number of signals is limited

doi:10.6342/NTU202401831

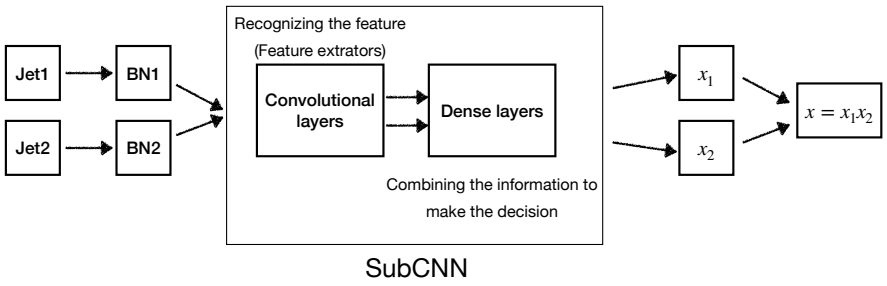


Figure 3.1: The plot of model architecture. The details of the subCNN is listed in Table. 3.1.

under weak supervision, the NN containing more parameters may not be trained effectively. Further optimization of the NN architecture is beyond the scope of this study.

Fig. 3.4 and Fig. 3.5 show the events score distribution after training for CWoLa with different significance before the NN cut. Two comments are in order. First, the distributions of signals (backgrounds) in both the training and testing datasets are similar. It shows that the NN truly discriminates the signals from backgrounds in both the training and testing phases under weak supervision. Second, the NN can give higher scores for signals but similar scores for backgrounds when training data contains more amounts of signals.

doi:10.6342/NTU202401831

Layers of CNN subnetwork	$\left(\begin{array}{l}\text{convolutional 2D layer: 64 filters with } 5\times 5 \text{ kernel size}\\ \text{maxpooling layer: } 2\times 2 \text{ pool size}\end{array}\right)\times 2$ convolutional 2D layer: 128 filters with 3×3 kernel size maxpooling layer: 2×2 pool size convolutional 2D layer: 128 filters with 3×3 kernel size flatten layer $(\text{dense layer: 128 units})\times 3$ dense layer (output): 1 unit
Layer setting	convolutional layer padding: same hidden layer activation function: ReLU output layer activation function: Sigmoid
Other	loss function: binary cross-entropy optimizer: Adam metric: accuracy batch size: 500 learning rate: 1e-3 (base learning, pretraining) learning rate: 1e-4 (CWoLa, fine-tuning, meta-learner updating) patience number: 20 (pretraining, meta-learning) patience number: 10 (CWoLa, fine-tuning)

Table 3.1: The CNN model subarchitecture and the hyperparameters

doi:10.6342/NTU202401831

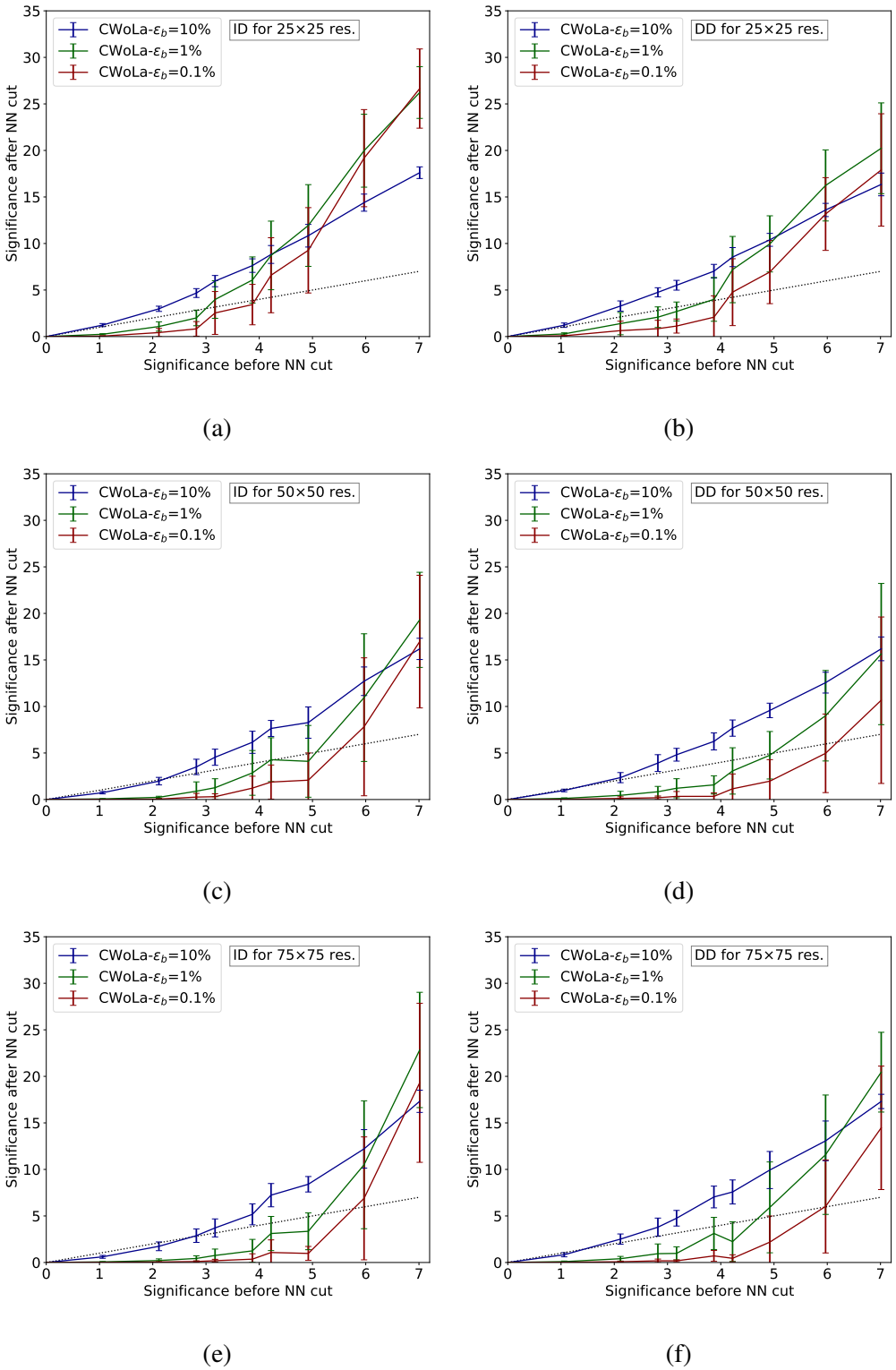
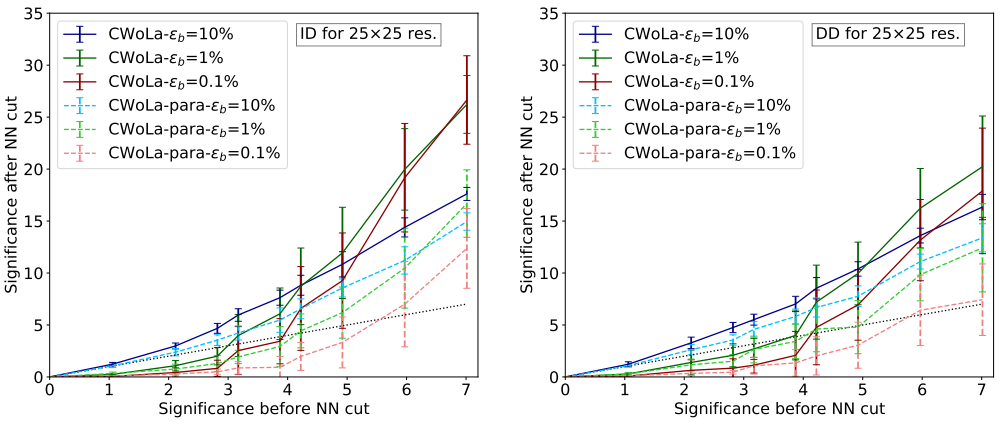


Figure 3.2: The results of CNN CWoLa for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25, 50×50 and 75×75 resolutions. The dotted line in each plot has a slope of 1.

doi:10.6342/NTU202401831



(a) (b)

Figure 3.3: The results of CNN CWoLa with different model architectures for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25. The dotted line in each plot has a slope of 1. The term *CWoLa* (solid lines) represents the NN containing a single subCNN, and the term *CWoLa-para* (dashed lines) represents the NN containing the two distinct subCNNs for two jet images. The dotted line in each plot has a slope of 1.

doi:10.6342/NTU202401831

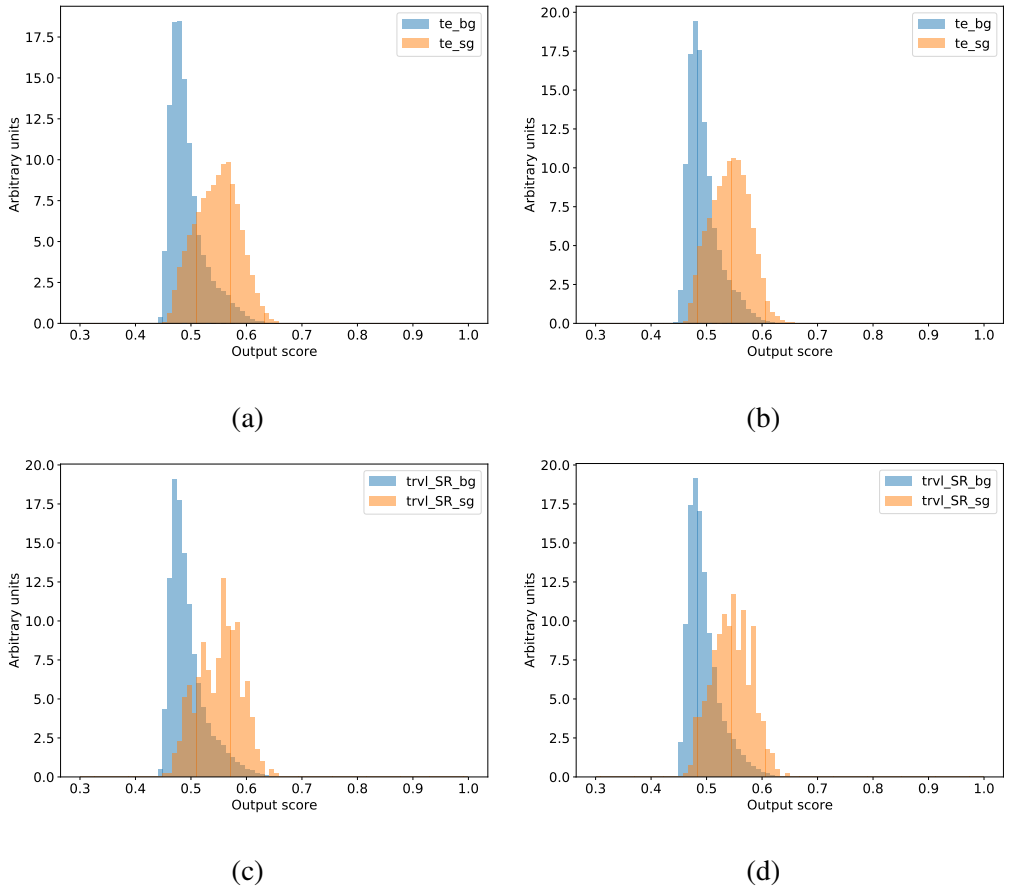


Figure 3.4: The score distributions of CNN CWoLa for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25 resolutions when the significance before the NN cut is 3.2. The terms *te_bg*, *te_sg*, *trvl_SR_bg*, and *trvl_SR_sg* are the testing background events in SR, testing signal events in SR, training background events in SR, and training signal events in SR. All distributions are normalized to unity.

doi:10.6342/NTU202401831

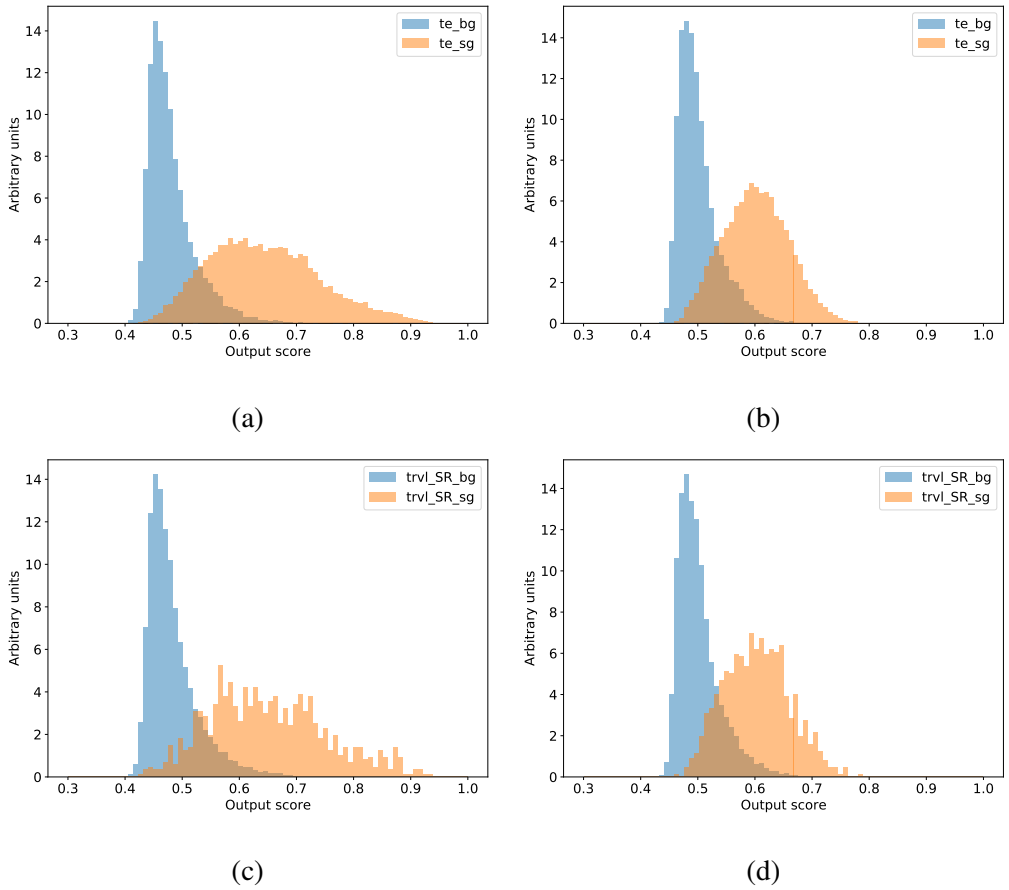


Figure 3.5: The score distributions of CNN CWoLa for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25 resolutions when the significance before the NN cut is 7.0. The terms *te_bg*, *te_sg*, *trvl_SR_bg*, and *trvl_SR_sg* are the testing background events in SR, testing signal events in SR, training background events in SR, and training signal events in SR. All distributions are normalized to unity.

doi:10.6342/NTU202401831

Chapter 4

Transfer learning

As illustrated in the previous chapter, the existence of a learning threshold makes the use of CWoLa problematic for small amounts of signal. A potential solution to this problem is transfer learning, which we introduce in this chapter.

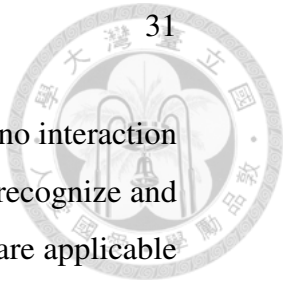
4.1 Introduction to transfer learning

The general concept of transfer learning involves having a neural network initially learn from a related problem with a large amount of data and then transfer some of this knowledge to the problem of interest. In practice, we use the techniques of pretraining and fine-tuning. Pretraining involves training NN parameters on a larger dataset, while fine-tuning refers to subsequent training on a smaller dataset. These larger and smaller datasets are referred to as the source and target data, respectively.

Two important remarks about transfer learning are in order:

1. Correlation between source and target tasks: In image classification problems, a high correlation between target tasks and source tasks is not always necessary. For instance, in Ref. [20], the pretrained model chosen was ResNet18 [57], which was initially trained on the ImageNet dataset [58]. This model was then fine-tuned with images of neutrino interactions, despite

doi:10.6342/NTU202401831



the fact that the images in ImageNet are not related to neutrino interaction images. In the pretraining phase, the feature extractors can recognize and distinguish geometric features (edges, corners, etc.), which are applicable to both source and target tasks. This demonstrates transfer learning can still work when source and target datasets are very different.

2. Finetuning strategies: The specific strategies for fine-tuning depend on various factors. Two common strategies are:

   (a) In the fine-tuning phase, NN models are trained with target data. However, training on target data may drastically change the parameters in the NN, causing an almost complete loss of the knowledge obtained from source data. This problem is called catastrophic forgetting [59]. Therefore, using a lower learning rate or freezing some of the layers (usually convolutional layers) in the NN can help avoid catastrophic forgetting. This strategy ensures that the NN can keep the knowledge from source data and reuse the knowledge when the NN trains with target data.

   (b) Adjusting dense layers: Due to differences between the source and target datasets or the architecture of the NN model, it may be necessary to replace the dense layers or randomly reinitialize the θ parameters in the dense layers. This allows the NN to adapt more easily to the target tasks.

   These strategies ensure that the NN can avoid catastrophic forgetting (by using lower learning rates or freezing layers) and that the NN can better adapt to the target tasks (by adjusting the dense layers or reinitializing θ).

doi:10.6342/NTU202401831

4.2 Implementation of Transfer Learning

In this study, rather than directly utilizing a pretrained model such as ResNet18, we pretrain our model on signals similar to the signal we will ultimately look for. The pretraining and fine-tuning strategies are implemented as follows:

First, the neural network is pretrained to distinguish a sample of pure background from a pure combination of different signals. This combination includes all the models mentioned in Chap. 2, except for the benchmark on which the model will be tested. In a real experiment, this would correspond to training on simulations. A total of 250k signal events and 250k background events from the signal region are used as the source data. It has been verified that increasing the size of the source data does not further improve performance. Four-fifths of the sample is used to update the NN parameters, and the remaining one-fifth is reserved for validation to prevent overfitting, both performed on pure samples.

Second, the neural network is fine-tuned to distinguish the pseudo-experiment data, mentioned in Sec. 3.2, i.e., the SR and SBs with the target benchmark signal mixed within the background. In a real experiment, this would represent fine-tuning on the actual data. The parameters of the feature extractor, denoted as Θ, are initialized with the values learned during pretraining, while the parameters of the dense layers, denoted as θ, are initialized randomly. During the fine-tuning step, Θ are frozen, and only θ are trained. A summary of the strategy is provided in Table 4.1.

Several comments on our strategies are in order. First, to fairly and reasonably compare the results of transfer learning and CWoLa, the choice of signal benchmarks in pretraining should not be the same as the target benchmark, so the NN will not directly learn the properties of target signals in advance. Second, to make sure that the NN can learn sufficient knowledge in the pre-training phase, the range of benchmarks in pretraining should be as large as possible to cover the properties of the target benchmark. Hence, the NN can reuse the necessary knowledge and train with target data. The HV module plays an ideal role in conveniently generating dif-

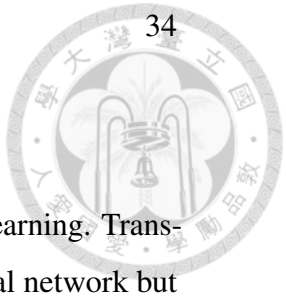
doi:10.6342/NTU202401831

	training set: (N_S, N_B)	Θ	θ	$(S, \overline{S})$
TL-pretraining	(250k, 250k) in the SR	Train	Train	-
TL-finetuning	pseudo-experiment data	Freeze	RI and train	-
MTL-pretraining	(250k, 250k) in the SR	Train	Train	Freeze
MTL-base learning	(2.5k, 2.5k)$\times$13 in the SR	Freeze	Train	Freeze
MTL-meta-learner	(2.5k, 2.5k)$\times$13 in the SR	Freeze	Train	Train
MTL-finetuning	pseudo-experiment data	Freeze	RI and train	Freeze

Table 4.1: The strategies summary for TL and MTL. For the pretraining phase for both TL and MTL, the signals of the training set contain all benchmarks listed in 2.1 except for the target benchmark used in the finetuning phase. In MTL, the base learning and meta-learning phases also use signal benchmarks from Table 2.1, excluding the target benchmark, forming 13 meta-tasks for base learning and meta-learner. The term *RI* means randomly initializing neural network parameters. For all phases except pretraining, the NN parameters will be initialized with values learned during previous steps unless specified by RI. All training sets, except those used in fine-tuning with pseudo-experiment data, are under full supervision with signals labeled as 1 and backgrounds as 0.

ferent benchmarks to enlarge the range of signals. Third, to make pretraining more effective, the pretraining phase is conducted under full supervision. This allows the NN to more easily distinguish the differences in properties between source signals and backgrounds and obtain better feature extractors during pretraining. Finally, since the NN is under full supervision during the pretraining phase and weak supervision during the fine-tuning phase, randomly initializing θ in the dense layers after pretraining is necessary. As discussed in Ref. [60], NN parameters in the deeper layers, which are closer to the output layers, are more class-specific, so randomly initializing θ can help the fine-tuning phase if the source and target tasks are considerably different.

doi:10.6342/NTU202401831



4.3 Discussion

Figure 4.1 compares the performance of pure CWoLa and transfer learning. Transfer learning not only enhances the overall performance of the neural network but also significantly lowers the learning threshold across all three resolutions for two target benchmarks. This means that the amount of signal needed to achieve a 5σ discovery is reduced by several times, as the neural network can more effectively identify signals and suppress background noise. Additionally, relative fluctuations in significance are minimized due to fewer trainable parameters and more effective learning.

Figure 4.2 compares the performance of transfer learning using different strategies with 25×25 resolutions. The results show that randomly initializing parameters θ in the dense layers is necessary and explain that θ parameters are more class-specific. While an optimal strategy can further enhance performance, both transfer learning results outperform CWoLa significantly.

Figures 4.3 and 4.4 display the event score distributions after fine-tuning with varying amounts of signals. Compared to Figures 3.4 and 3.5, the neural network successfully assigns higher scores to signal events even with limited amounts of signals. Additionally, the distributions of signals (backgrounds) in both training and testing datasets are similar, as observed in Sec. 3.3.

In summary, these results demonstrate that transfer learning significantly improves the neural network's ability to distinguish signal from background, even with limited signal data. Pretraining enables the neural network to develop better feature extractors, facilitating faster and more effective learning during the fine-tuning phase. The improved performance is evident in both the overall significance for different benchmark models and resolutions.

doi:10.6342/NTU202401831

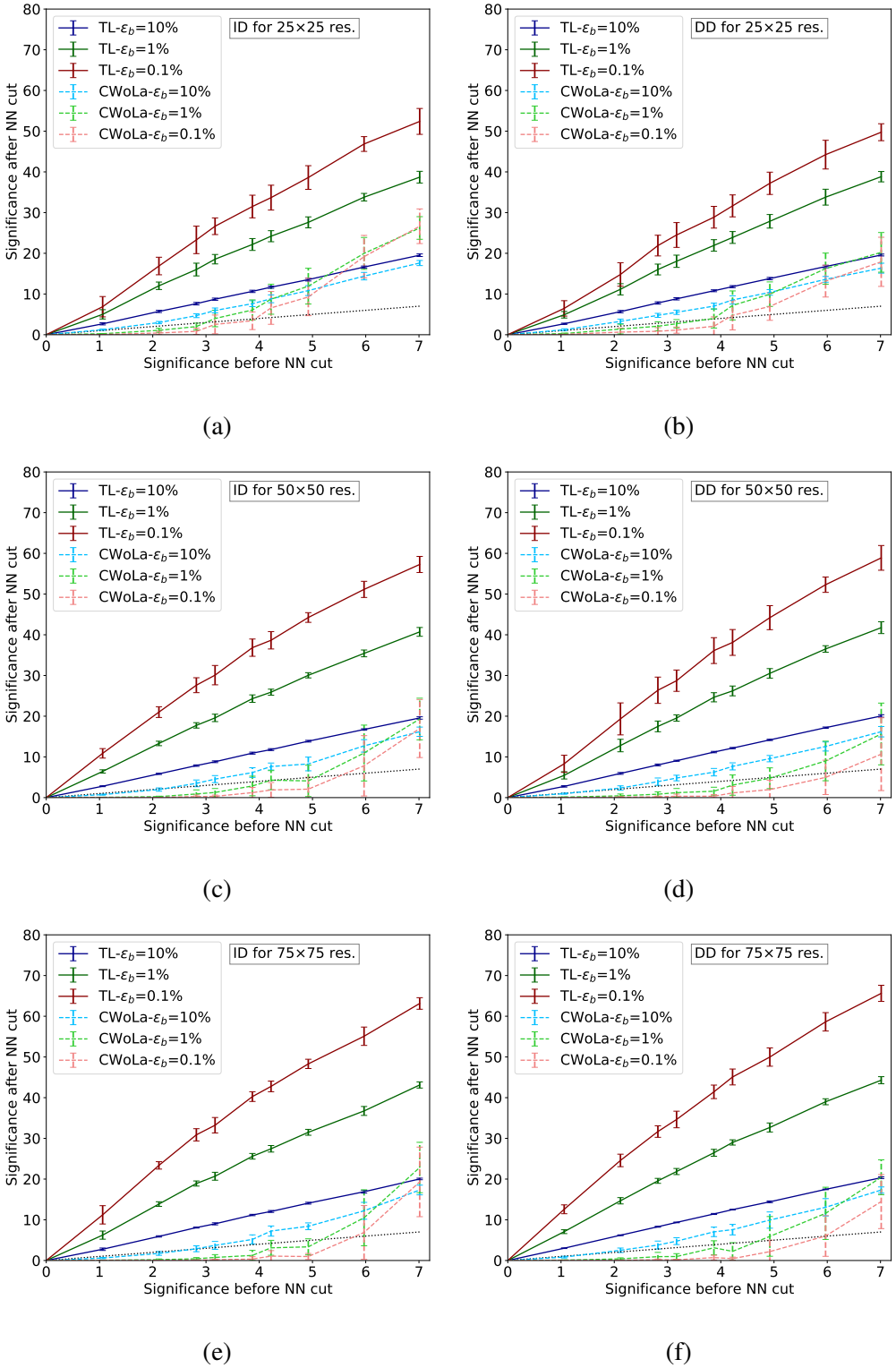
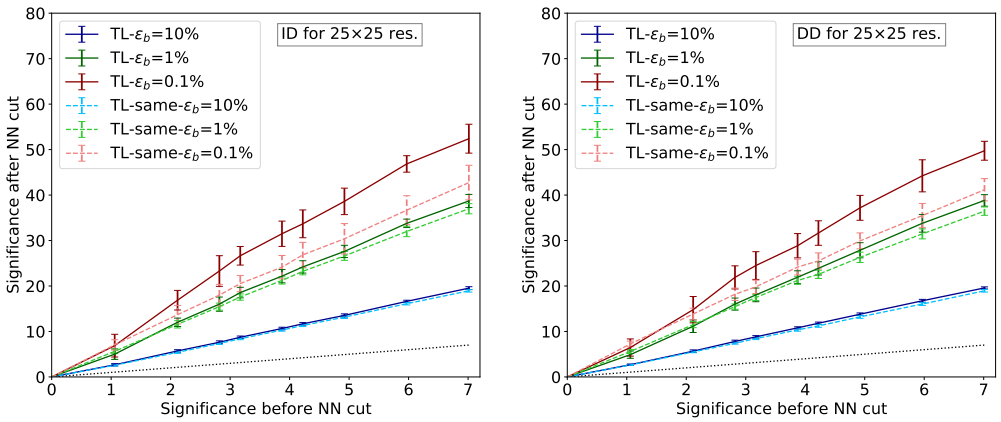


Figure 4.1: The results of transfer learning (solid curves) and of CWoLa (dashed curves, same as those in Fig. 3.2) for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25, 50×50 and 75×75 resolutions. The dotted line in each plot has a slope of 1.

doi:10.6342/NTU202401831



(a) (b)

Figure 4.2: The results of CNN transfer learning with different finetuning strategies for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25. The dotted line in each plot has a slope of 1. The term *TL* (solid lines) represents the strategy mentioned in the text, and the term *TL-same* (dashed lines) represents the the strategy without the randomly initializing the θ in dense layers after pretraining. The dotted line in each plot has a slope of 1.

doi:10.6342/NTU202401831

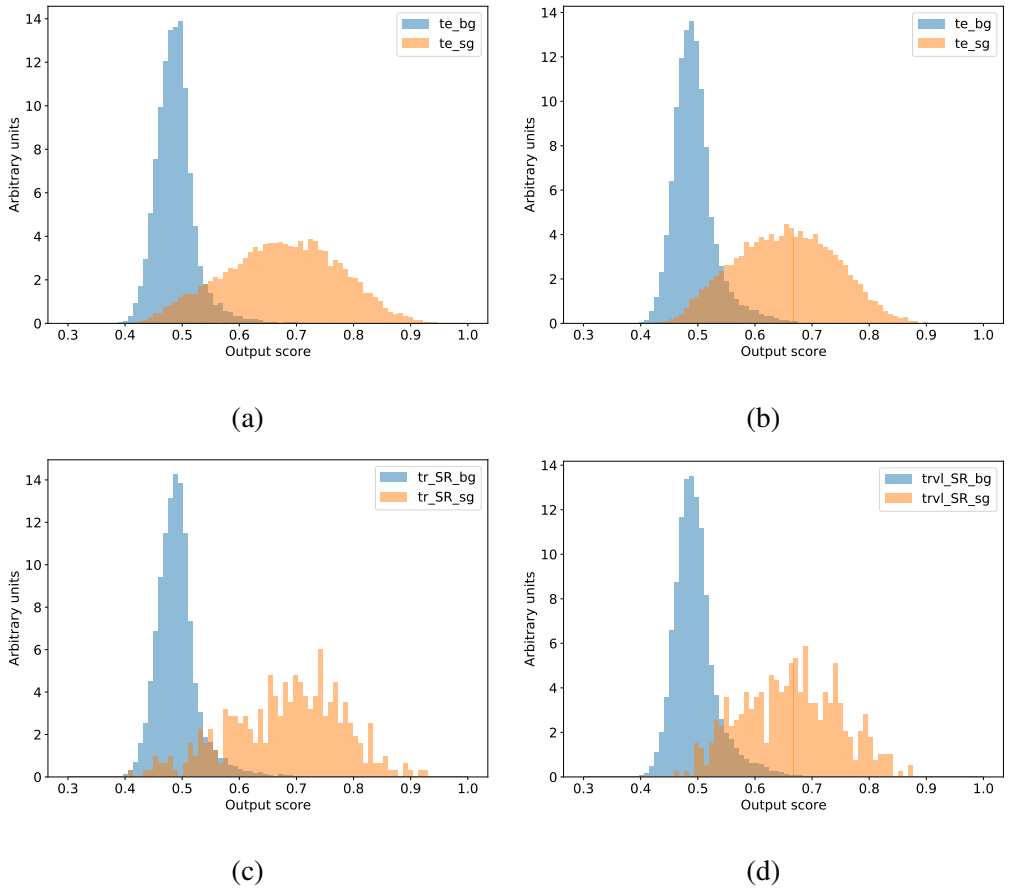


Figure 4.3: The score distributions of CNN transfer learning for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25 resolutions when the significance before the NN cut is 3.2. The terms *te_bg*, *te_sg*, *trvl_SR_bg*, and *trvl_SR_sg* are the testing background events in SR, testing signal events in SR, training background events in SR, and training signal events in SR. All distributions are normalized to unity.

doi:10.6342/NTU202401831

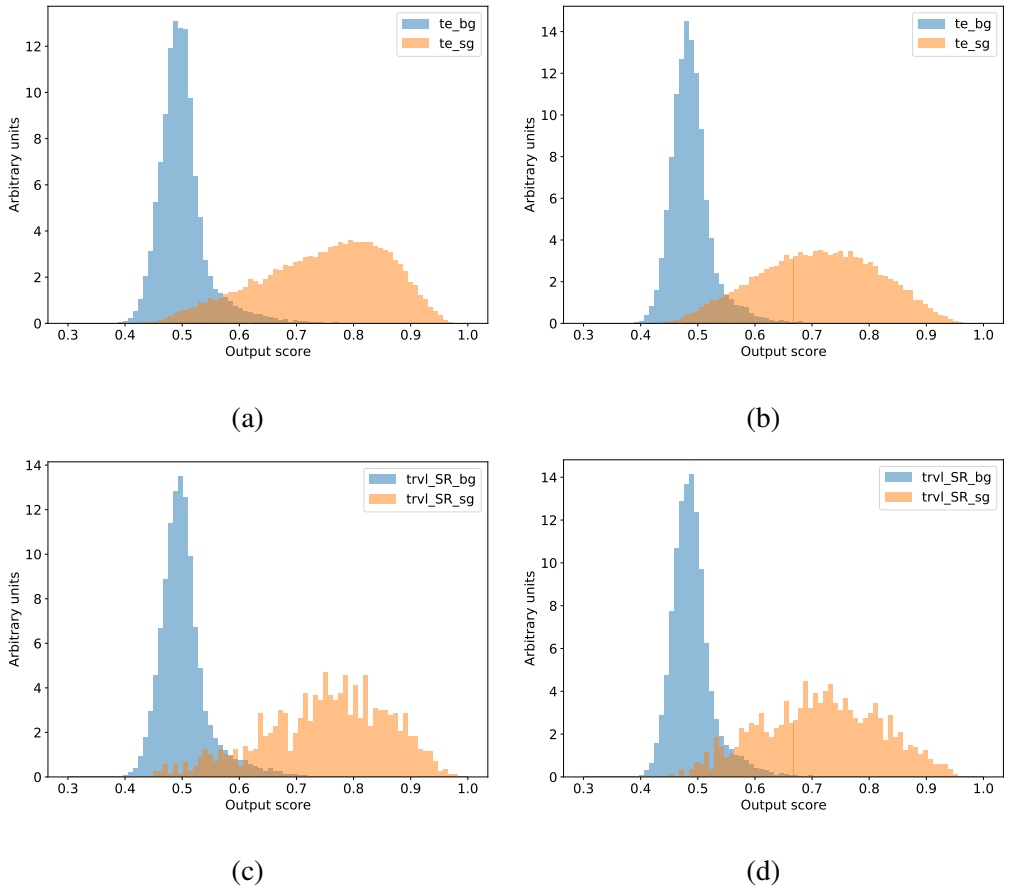


Figure 4.4: The score distributions of CNN transfer learning for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25 resolutions when the significance before the cut is 7.0. The terms *te_bg*, *te_sg*, *trvl_SR_bg*, and *trvl_SR_sg* are the testing background events in SR, testing signal events in SR, training background events in SR, and training signal events in SR. All distributions are normalized to unity.

doi:10.6342/NTU202401831

Chapter 5

Meta learning

5.1 Introduction to meta-transfer learning

Meta-learning is an alternative approach for creating neural networks that can learn from less data. The general idea is not to reuse concepts from related tasks but rather to teach the neural network how to learn tasks more efficiently. Specifically, we study the use of meta-transfer learning (MTL) [61]. Although many other techniques exist, we choose MTL because it is closely related to transfer learning, which has already been shown to be very successful in the previous chapter. We will present our implementation of MTL, which we simplify and modify somewhat, and refer to Ref. [61] for more details.

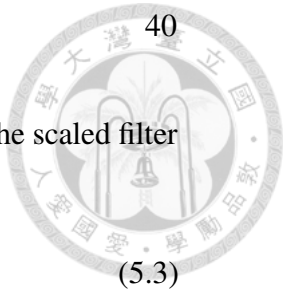
MTL utilizes scaling and shifting parameters to enhance learning efficiency. Consider a rectangular image A of arbitrary dimensions and M channels, with a set of N convolutional filters previously created. The filters and their indices are labeled as

$$F_{ij}^{cf}, \tag{5.1}$$

where the index f refers to the label of the filter (running from 1 to N), i and j correspond to the positional arguments of the filter (η and ϕ in our case), and c corresponds to the channel (running from 1 to M). Scaling is applied as

$$\bar{F}_{ij}^{cf} = S^{cf} F_{ij}^{cf}, \tag{5.2}$$

doi:10.6342/NTU202401831



where S^{cf} are the scaling parameters and $\bar{F}^f$ are the scaled filters. The scaled filter $\bar{F}^f$ is then applied to image A at point (i, j) as

$$B^f_{i'j'} = g\left((\bar{F}^f \star A)_{ij} + b^f + \bar{S}^f\right), \tag{5.3}$$

where B is the resulting image, g is the activation function, $\star$ is the cross-correlation operation, b^f are the previously determined biases, and $\bar{S}^f$ are the shifting parameters. The indices i' and j' are related to the positions i and j, though the exact relation depends on other parameters (stride, padding, etc.). The scaling and shifting parameters are optimized to make the neural network learn faster and are meant to emphasize more important features. These parameters are crucial to how the neural network "learns-to-learn".

5.2 Implementation of meta-transfer learning

The architecture of our neural network remains mostly identical to Table 3.1. The only modification is the inclusion of scaling and shifting parameters in all convolutional layers. As before, the NN parameters of the feature extractor are denoted as Θ and those of the dense layers as θ. The training proceeds in three phases.

First, pretraining is conducted as described in Sec. 4.2. During this phase, the neural network learns to distinguish between background samples and a mixture of different signals except the benchmark used in pseudo-experiment data. The scaling parameters and shifting parameters are kept at 1 and 0, respectively. After completing the pretraining, the NN model parameters Θ are fixed permanently. However, unlike the method in Ref. [61], the θ parameters are not initialized randomly and this way will obtain better results in our case.

Second, a new phase called meta-training is performed. Consider a series of tasks $\mathcal{T}$ forming a task-space $p(\mathcal{T})$. For our purposes, the tasks correspond to different models from Sec. 2.1, excluding the benchmark used for pseudo-experiment data under weak supervision. The training is schematically represented

doi:10.6342/NTU202401831

as follows:

for `episode` **do**
  for $\mathcal{T}$ `in` $p(\mathcal{T})$ **do**
    `base learning`
    `meta-learner update`
    `evaluation of` $\mathcal{L}_{\mathcal{T}}$
  end for
  `average` $\mathcal{L}_{\mathcal{T}}$ `over` $p(\mathcal{T})$
  `test for early stopping`
end for

In detail, an episode in meta-learning is equivalent to an epoch in the regular NN training. Each possible task in the task-space is considered once per episode. The first step of each episode involves an inner-loop where the following steps are executed for each task in the task-space:

- `base learning`: A series of temporary θ parameters labelled as θ' are obtained via gradient descent as

$$\theta' \leftarrow \theta - \beta \nabla_{\theta} \mathcal{L}_{\mathcal{T}}(\Theta, \theta, S, \bar{S}), \tag{5.4}$$

  where β is the learning rate in the base learning step and $\mathcal{L}_{\mathcal{T}}$ the loss function. The training is performed over only 3 epochs to prevent overfitting.

- `meta-learner update`: The θ, scaling S and shifting $\bar{S}$ parameters are updated by one step of gradient descent as

$$\begin{aligned} \theta &=: \theta - \gamma \nabla_{\theta} \mathcal{L}_{\mathcal{T}}(\Theta, \theta', S, \bar{S}), \\ S &=: S - \gamma \nabla_{S} \mathcal{L}_{\mathcal{T}}(\Theta, \theta', S, \bar{S}), \\ \bar{S} &=: \bar{S} - \gamma \nabla_{\bar{S}} \mathcal{L}_{\mathcal{T}}(\Theta, \theta', S, \bar{S}), \end{aligned} \tag{5.5}$$

  where γ is the learning rate in the meta-learner updating step. After completing this step, the temporary parameters θ' will not be used anymore and can be discarded.

doi:10.6342/NTU202401831

- `evaluation of` $\mathcal{L}_{\mathcal{T}}$: The loss function is evaluated using the updated parameters: $\mathcal{L}_{\mathcal{T}}(\Theta, \theta, S, \bar{S})$. This will be used to determine when to stop meta-training.

During the base learning and meta-learner update, the NN is trained to distinguish pure samples of 2.5k signals and 2.5k backgrounds in the SR. Four-fifths of the sample is used for training and the other one-fifth of the sample is used for validation. Training is done under full supervision. Different events are used for each of the three steps in the inner-loop of each episode. Once the inner-loop is complete, the $\mathcal{L}_{\mathcal{T}}$ are averaged and used to test for early stopping. After completing the whole meta-training phase, the θ parameters are initialized randomly.

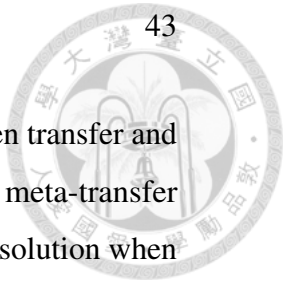
Third, fine-tuning is performed similarly to Sec. 4.2, with the difference being the presence of scaling and shifting parameters learned during meta-training but kept fixed in this phase.

Note that our method is simplified compared to the original method in Ref. [61]. The primary difference is that we omitted the hard tasks algorithm, as it was beyond the scope of this initial study on the applicability of meta-learning to CWoLa. Additionally, we did not implement meta-batches, the meta-learning equivalent of a batch, as they were mostly irrelevant without the hard tasks algorithm. A summary of the strategy is provided in Table 4.1.

5.3 Discussion

Fig. 5.1 shows the comparison between transfer learning and meta-transfer learning. Meta-transfer learning generally exhibits a slight performance improvement for the 25×25 and 50×50 resolutions compared to transfer learning, attributed to the additional adjustments provided by the scaling and shifting parameters. It is important to note that the results for transfer learning are already close to the mathematical upper limits, leaving limited room for further improvement at high significance levels. However, the relative improvement at low significance levels

doi:10.6342/NTU202401831



can be substantial. For the 75×75 resolution, the difference between transfer and meta-transfer learning is negligible. Nevertheless, we observe that meta-transfer learning can slightly outperform transfer learning for the 75×75 resolution when a larger kernel size is employed, as illustrated in Fig. 5.2. A comprehensive study on kernel size optimization is beyond the scope of this work.

Fig. 5.3, Fig. 5.4 and Fig. 5.5 show the distributions of the scaling and shifting parameters after meta-learning. Obviously, the scaling and shifting parameters provide a minor adjustment for feature extractors Θ. For the higher resolutions, the distributions of the scaling and shifting are more centralized at 1 and 0, respectively. Fig. 5.6 shows the distributions with the larger kernel size. With larger sizes of the kernel, the distributions are more spread out to provide relatively useful adjustments to feature extractors.

doi:10.6342/NTU202401831

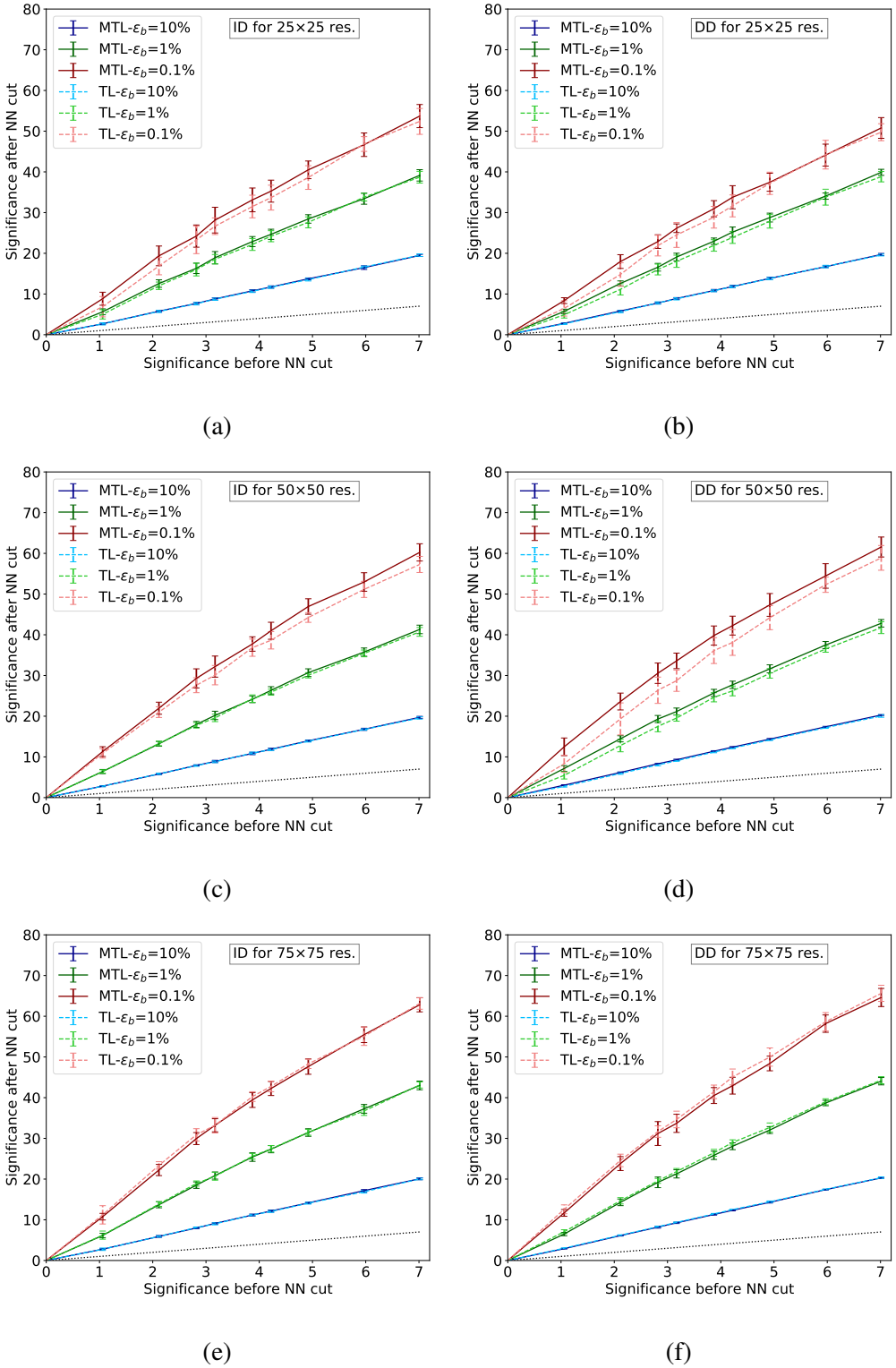
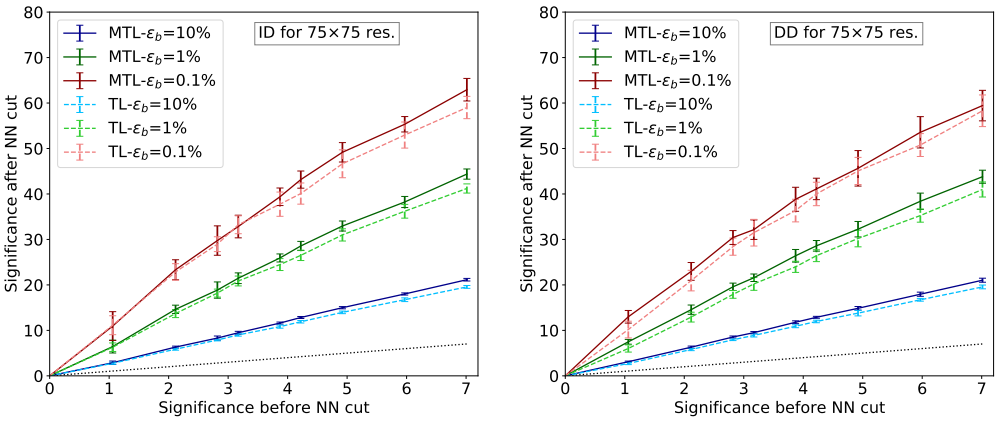


Figure 5.1: The results of meta-transfer learning (solid curves) and transfer learning (dashed curves, same as those in Fig. 4.1) for the ID (left column) and DD (right column) scenarios with Λ_D= 10 GeV for 25×25, 50×50 and 75×75 resolutions. The dotted line in each plot has a slope of 1.

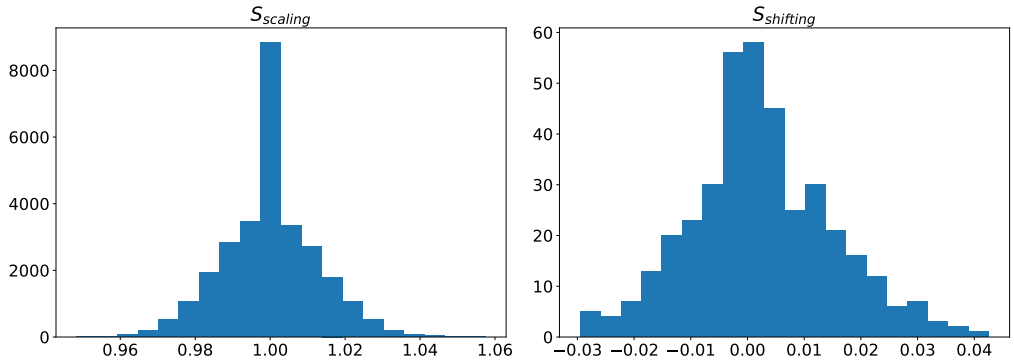
doi:10.6342/NTU202401831



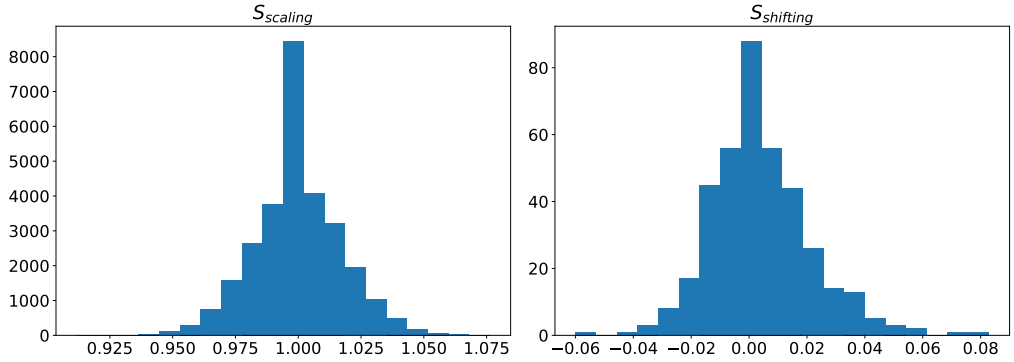
(a) (b)

Figure 5.2: The results of meta-transfer learning (solid curves) and transfer learning (dashed curves) for the ID (left) and DD (right) scenarios with Λ_D= 10 GeV for 75×75 resolution with a larger size of kernels. The kernel sizes are 10×10 and 5×5 respectively instead of 5×5 and 3×3 mentioned in Table 3.1. The dotted line in each plot has a slope of 1.

doi:10.6342/NTU202401831



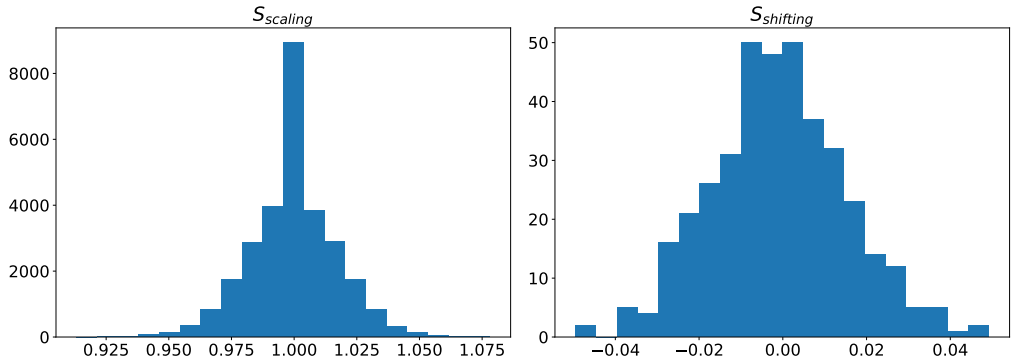
(a)



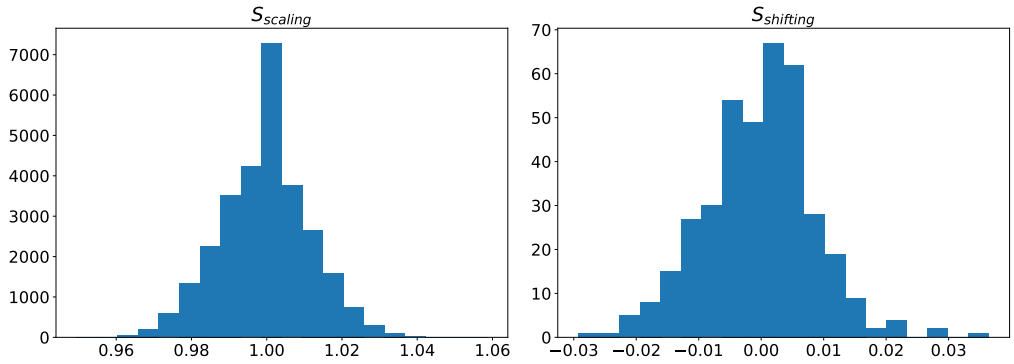
(b)

Figure 5.3: The distributions of scaling (left) and shifting (right) parameters $S, \bar{S}$ for ID (upper) and DD (lower) scenarios with Λ_D=10 GeV for 25×25 resolution.

doi:10.6342/NTU202401831



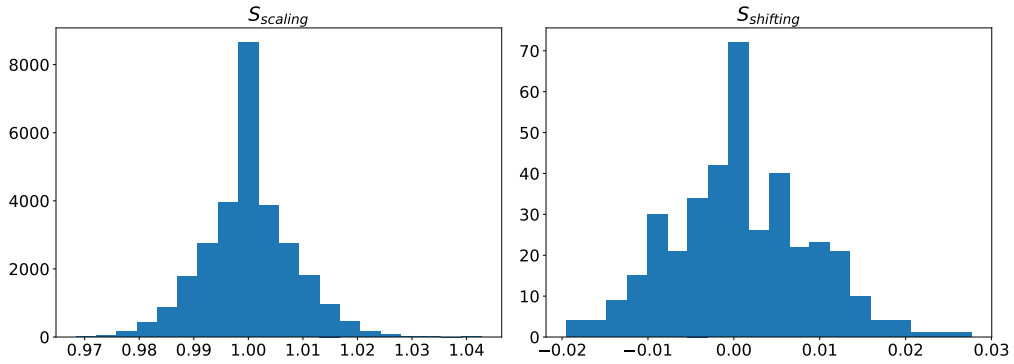
(a)



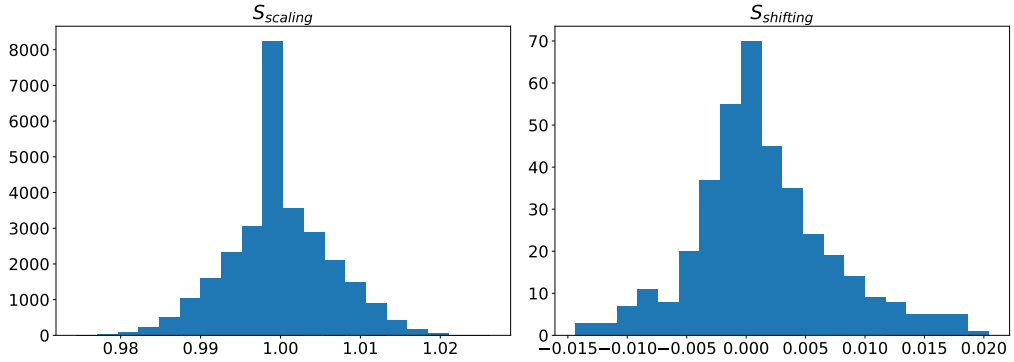
(b)

Figure 5.4: The distributions of scaling (left) and shifting (right) parameters $S, \bar{S}$ for ID (upper) and DD (lower) scenarios with Λ_D=10 GeV for 50×50 resolution.

doi:10.6342/NTU202401831



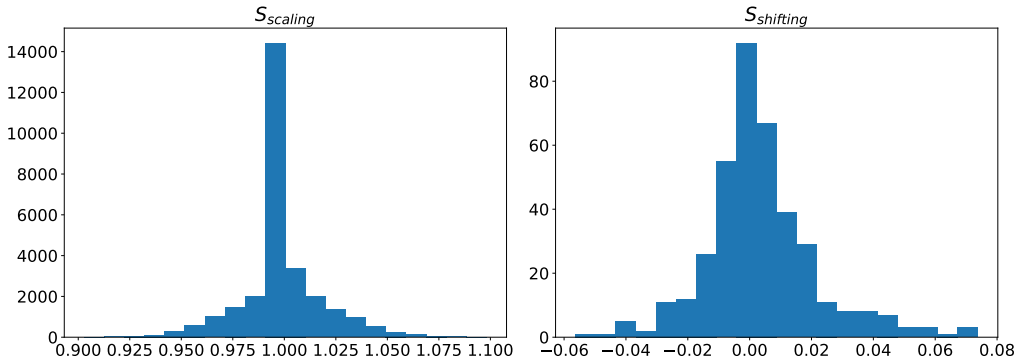
(a)



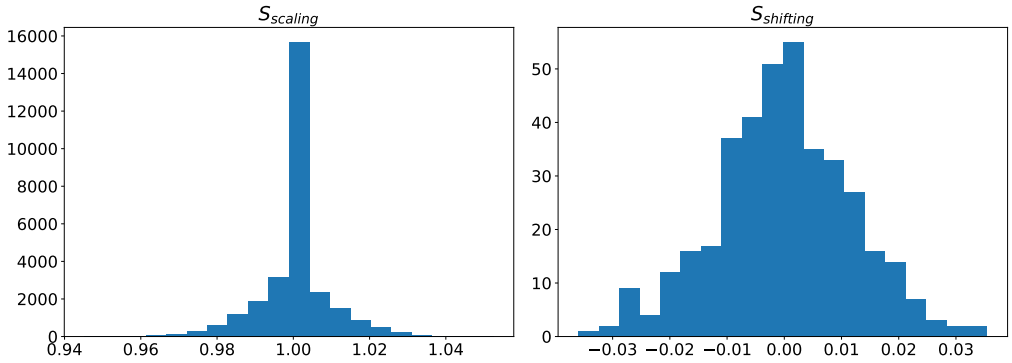
(b)

Figure 5.5: The distributions of scaling (left) and shifting (right) parameters $S, \bar{S}$ for ID (upper) and DD (lower) scenarios with Λ_D=10 GeV for 75×75 resolution.

doi:10.6342/NTU202401831



(a)



(b)

Figure 5.6: The distributions of scaling (left) and shifting (right) parameters $S, \bar{S}$ for ID (upper) and DD (lower) scenarios with Λ_D=10 GeV for 75×75 resolution with a larger size of kernels. The kernel sizes are 10×10 and 5×5 respectively instead of 5×5 and 3×3 mentioned in Table 3.1..

doi:10.6342/NTU202401831

Chapter 6

Conclusion

Weak supervision searches offer the dual advantages of being able to train on real data and exploiting distinctive signal properties. However, training a neural network via weak supervision often demands an impractically large amount of signal, nearly to the extent that the signal could have been discovered without the neural network. To address this issue, our work focuses on developing neural networks that can learn from less signal using transfer and meta-learning. The primary idea is to first train a neural network on simulations, enabling it to learn relevant concepts or become a more efficient learner. Subsequently, the neural network is trained on experimental data, requiring less signal due to its previous training. Our implementation of this procedure involves transfer learning and meta-transfer learning.

We find that transfer learning significantly enhances the performance of CWoLa searches. This improvement is particularly notable at low significance, reducing the amount of signal needed for discovery by a substantial factor. Meta-transfer learning further enhances CWoLa searches, though not dramatically.

We emphasize that this work serves as a proof of principle, and several questions remain unanswered. Specifically, the choice of models for training may influence the ability to discover signals that differ significantly from them. The extent of this effect is left for future investigation. However, a small reduction to the scope of

doi:10.6342/NTU202401831

model sensitivity seems a fair prize to pay for the magnitude of our improvement over the regular CWoLa method.

After publication of our work, Ref. [13] by Cheng, Singh and Nachman proposed a search strategy similar to ours which they dubbed Prior-Assisted Weak Supervision (PAWS). Although differing in some details, PAWS also consists of pretraining on simulations and performing weak supervision on actual data. They showed that the combination of pretraining and weak supervision could improve the sensitivity of searches by a factor of ~ 10. Their figure 2 bears striking similarity with some of our results. Ref. [14] also studied the combination of pretraining on simulations and weak supervision on data, their technique Sophon. They claim their method can improve signal sensitivity by a factor of a few.

Finally, it is important to note that transfer and meta-learning are extensive and rapidly evolving fields. Although we demonstrated their potential, we only explored two specific techniques. It is likely that more powerful techniques already exist or could be developed in the future. Additionally, we did not fully optimize our analysis, and there are clear opportunities for refinement. Given our promising results, we believe further studies on transfer and meta-learning and developing other techniques for weak supervision are highly warranted.

doi:10.6342/NTU202401831

Reference

[1] H. Beauchesne, Z.-E. Chen and C.-W. Chiang, *Improving the performance of weak supervision searches using transfer and meta-learning*, *JHEP* **02** (2024) 138 [2312.06152]. i

[2] Particle Data Group collaboration, *Review of Particle Physics*, *PTEP* **2022** (2022) 083C01. 1

[3] Y.-C.J. Chen, C.-W. Chiang, G. Cottin and D. Shih, *Boosted W and Z tagging with jet charge and deep learning*, *Phys. Rev. D* **101** (2020) 053001 [1908.08256]. 2

[4] E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer and A. Mück, *Casting a graph net to catch dark showers*, *SciPost Phys.* **10** (2021) 046 [2006.08639]. 2

[5] S. Chang, T.-K. Chen and C.-W. Chiang, *Distinguishing W' signals at hadron colliders using neural networks*, *Phys. Rev. D* **103** (2021) 036016 [2007.14586]. 2

[6] C.-W. Chiang, D. Shih and S.-F. Wei, *VBF vs. GGF Higgs with Full-Event Deep Learning: Towards a Decay-Agnostic Tagger*, *Phys. Rev. D* **107** (2023) 016014 [2209.05518]. 2

[7] E.M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: Learning from mixed samples in high energy physics*, *JHEP* **10** (2017) 174 [1708.02949]. 2, 3, 13, 20

doi:10.6342/NTU202401831

[8] M. Farina, Y. Nakai and D. Shih, *Searching for New Physics with Deep Autoencoders*, *Phys. Rev. D* **101** (2020) 075021 [1808.08992]. 3

[9] J. Batson, C.G. Haaf, Y. Kahn and D.A. Roberts, *Topological Obstructions to Autoencoding*, *JHEP* **04** (2021) 280 [2102.08380]. 3

[10] B.M. Dillon, T. Plehn, C. Sauer and P. Sorrenson, *Better Latent Spaces for Better Autoencoders*, *SciPost Phys.* **11** (2021) 061 [2104.08291]. 3

[11] ATLAS collaboration, *Dijet resonance search with weak supervision using $\sqrt{s} = 13$ TeV pp collisions in the ATLAS detector*, *Phys. Rev. Lett.* **125** (2020) 131801 [2005.02983]. 4

[12] J.H. Collins, P. Martín-Ramiro, B. Nachman and D. Shih, *Comparing weak- and unsupervised methods for resonant anomaly detection*, *Eur. Phys. J. C* **81** (2021) 617 [2104.02092]. 4

[13] C.L. Cheng, G. Singh and B. Nachman, *Incorporating Physical Priors into Weakly-Supervised Anomaly Detection*, 2405.08889. 4, 5, 51

[14] C. Li et al., *Accelerating Resonance Searches via Signature-Oriented Pre-training*, 2405.12972. 4, 5, 51

[15] B.M. Dillon, L. Favaro, F. Feiden, T. Modak and T. Plehn, *Anomalies, Representations, and Self-Supervision*, 2301.04660. 4

[16] J.H. Collins, K. Howe and B. Nachman, *Anomaly Detection for Resonant New Physics with Machine Learning*, *Phys. Rev. Lett.* **121** (2018) 241803 [1805.02664]. 4, 22

[17] T. Finke, M. Hein, G. Kasieczka, M. Krämer, A. Mück, P. Prangchaikul et al., *Back To The Roots: Tree-Based Algorithms for Weakly Supervised Anomaly Detection*, 2309.13111. 4

doi:10.6342/NTU202401831

[18] M. Freytsis, M. Perelstein and Y.C. San, *Anomaly Detection in Presence of Irrelevant Features*, 2310.13057. 4

[19] F.A. Dreyer, R. Grabarczyk and P.F. Monni, *Leveraging universality of jet taggers through transfer learning*, *Eur. Phys. J. C* **82** (2022) 564 [2203.06210]. 5

[20] A. Chappell and L.H. Whitehead, *Application of transfer learning to neutrino interaction classification*, *Eur. Phys. J. C* **82** (2022) 1099 [2207.03139]. 5, 30

[21] M.P. Kuchera, R. Ramanujan, J.Z. Taylor, R.R. Strauss, D. Bazin, J. Bradt et al., *Machine Learning Methods for Track Classification in the AT-TPC*, *Nucl. Instrum. Meth. A* **940** (2019) 156 [1810.10350]. 5

[22] W. Wei et al., *Deep transfer learning for star cluster classification: I. application to the PHANGS–HST survey*, *Mon. Not. Roy. Astron. Soc.* **493** (2020) 3178 [1909.02024]. 5

[23] M. Eriksen et al., *The PAU Survey: Photometric redshifts using transfer learning from simulations*, *Mon. Not. Roy. Astron. Soc.* **497** (2020) 4565 [2004.07979]. 5

[24] D. George, H. Shen and E.A. Huerta, *Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO*, 1706.07446. 5

[25] D. George, H. Shen and E.A. Huerta, *Glitch Classification and Clustering for LIGO with Deep Transfer Learning*, in *NiPS Summer School 2017*, 11, 2017 [1711.07468]. 5

[26] T. Kishimoto, M. Morinaga, M. Saito and J. Tanaka, *Application of transfer learning to event classification in collider physics*, *PoS* **ISGC2022** (2022) 016. 5

doi:10.6342/NTU202401831

[27] M.J. Dolan and A. Ore, *Metalearning and data augmentation for mass-generalized jet taggers*, *Phys. Rev. D* **105** (2022) 094030 [2111.06047]. 5

[28] T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey, *Meta-learning in neural networks: A survey*, *IEEE Transactions on Pattern Analysis; Machine Intelligence* **44** (2022) 5149. 5

[29] L. Carloni, J. Rathsman and T. Sjostrand, *Discerning Secluded Sector gauge structures*, *JHEP* **04** (2011) 091 [1102.3795]. 7

[30] L. Carloni and T. Sjostrand, *Visible Effects of Invisible Hidden Valley Radiation*, *JHEP* **09** (2010) 105 [1006.2911]. 7

[31] G. Albouy et al., *Theory, phenomenology, and experimental avenues for dark showers: a Snowmass 2021 report*, *Eur. Phys. J. C* **82** (2022) 1132 [2203.09503]. 7, 8

[32] H. Beauchesne, E. Bertuzzo and G. Grilli Di Cortona, *Dark matter in Hidden Valley models with stable and unstable light dark mesons*, *JHEP* **04** (2019) 118 [1809.10152]. 7

[33] E. Bernreuther, F. Kahlhoefer, M. Krämer and P. Tunney, *Strongly interacting dark sectors in the early Universe and at the LHC through a simplified portal*, *JHEP* **01** (2020) 162 [1907.04346]. 7

[34] H. Beauchesne and G. Grilli di Cortona, *Classification of dark pion multiplets as dark matter candidates and collider phenomenology*, *JHEP* **02** (2020) 196 [1910.10724]. 7

[35] CMS collaboration, *Search for new particles decaying to a jet and an emerging jet*, *JHEP* **02** (2019) 179 [1810.10069]. 7

doi:10.6342/NTU202401831

[36] CMS collaboration, *Search for resonant production of strongly coupled dark matter in proton-proton collisions at 13 TeV*, *JHEP* **06** (2022) 156 [2112.11125]. 7

[37] ATLAS collaboration, *Search for non-resonant production of semi-visible jets using Run 2 data in ATLAS*, *Phys. Lett. B* **848** (2024) 138324 [2305.18037]. 7

[38] ATLAS collaboration, *Search for Resonant Production of Dark Quarks in the Dijet Final State with the ATLAS Detector*, 2311.03944. 7

[39] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079 [1405.0301]. 9

[40] R.D. Ball et al., *Parton distributions with LHC data*, *Nucl. Phys. B* **867** (2013) 244 [1207.1303]. 9

[41] DELPHES 3 collaboration, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [1307.6346]. 9

[42] M. Cacciari, G.P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896 [1111.6097]. 9

[43] A.J. Larkoski, J. Thaler and W.J. Waalewijn, *Gaining (Mutual) Information about Quark/Gluon Discrimination*, *JHEP* **11** (2014) 129 [1408.3122]. 12

[44] C.F. Berger, T. Kucs and G.F. Sterman, *Event shape / energy flow correlations*, *Phys. Rev. D* **68** (2003) 014012 [hep-ph/0303051]. 12

[45] L.G. Almeida, S.J. Lee, G. Perez, G.F. Sterman, I. Sung and J. Virzi, *Substructure of high-p_T Jets at the LHC*, *Phys. Rev. D* **79** (2009) 074017 [0807.0234]. 12

doi:10.6342/NTU202401831

[46] S.D. Ellis, C.K. Vermilion, J.R. Walsh, A. Hornig and C. Lee, *Jet Shapes and Jet Algorithms in SCET*, *JHEP* **11** (2010) 101 [1001.0014]. 12

[47] CMS collaboration, *Search for a Higgs Boson in the Decay Channel $H \to ZZ^* \to q\bar{q}\ell^-\ell^+$ in pp Collisions at $\sqrt{s} = 7$ TeV*, *JHEP* **04** (2012) 036 [1202.1416]. 13

[48] P. Gras, S. Höche, D. Kar, A. Larkoski, L. Lönnblad, S. Plätzer et al., *Systematics of quark/gluon tagging*, *JHEP* **07** (2017) 091 [1704.03878]. 13

[49] J.R. Andersen et al., *Les Houches 2015: Physics at TeV Colliders Standard Model Working Group Report*, in *9th Les Houches Workshop on Physics at TeV Colliders*, 5, 2016 [1605.04692]. 13

[50] A. Butter et al., *The Machine Learning landscape of top taggers*, *SciPost Phys.* **7** (2019) 014 [1902.09914]. 13

[51] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images — deep learning edition*, *JHEP* **07** (2016) 069 [1511.05190]. 13

[52] G. Kasieczka, T. Plehn, M. Russell and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, *JHEP* **05** (2017) 006 [1701.08784]. 13

[53] J. Neyman and E.S. Pearson, *On the Problem of the Most Efficient Tests of Statistical Hypotheses*, *Phil. Trans. Roy. Soc. Lond. A* **231** (1933) 289. 20

[54] F. Chollet et al., "Keras." https://keras.io, 2015. 22

[55] M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, 1603.04467. 22

[56] ATLAS collaboration, *Formulae for Estimating Significance*, 2020. 23

doi:10.6342/NTU202401831

[57] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 30

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma et al., *Imagenet large scale visual recognition challenge*, *International journal of computer vision* **115** (2015) 211. 30

[59] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu et al., *Overcoming catastrophic forgetting in neural networks*, *Proceedings of the national academy of sciences* **114** (2017) 3521. 31

[60] M.D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks*, in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833, Springer, 2014. 33

[61] Q. Sun, Y. Liu, T.-S. Chua and B. Schiele, *Meta-transfer learning for few-shot learning*, 1812.02391. 39, 40, 42

doi:10.6342/NTU202401831