Task Understanding from Confusing Multi-task Data

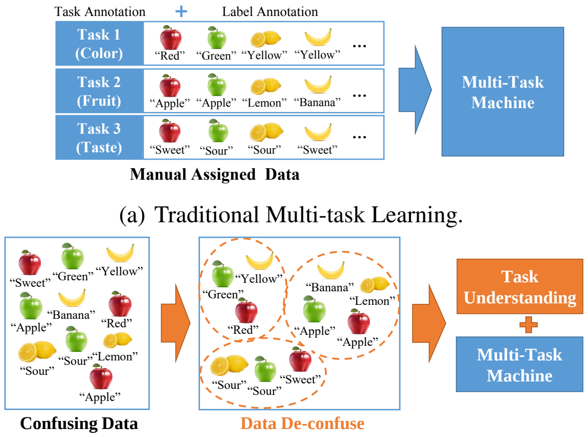
Xin Su [1 2] **Yizhou Jiang** [1] **Shangqi Guo** [1 3] **Feng Chen** [1 3 2]

Abstract

Beyond machine learning's success in the specific tasks, research for learning multiple tasks simultaneously is referred to as multi-task learning. However, existing multi-task learning needs manual definition of tasks and manual task annotation. A crucial problem for advanced intelligence is how to understand the human task concept using basic input-output pairs. Without task definition, samples from multiple tasks are mixed together and result in a confusing mapping challenge. We propose *Confusing Supervised Learning (CSL)* that takes these confusing samples and extracts task concepts by differentiating between these samples. We theoretically proved the feasibility of the CSL framework and designed an iterative algorithm to distinguish between tasks. The experiments demonstrate that our CSL methods could achieve a human-like task understanding without task labeling in multi-function regression problems and multi-task recognition problems.

1. Introduction

Over the past few decades, machine learning research has reached or even exceeded human-level performance on various problems (Silver et al., 2016; He et al., 2015). However, these learning machines are limited to a specific task in a determined environment, which is referred to as "Narrow AI" (Kurzweil, 2005). Beyond this paradigm, a type of intelligent system that processes multiple tasks simultaneously is referred to as a multi-task learning system (Caruana, 1997). The systems learn from labeled data referring to multiple tasks and give corresponding inferences from the input for every task. For instance, as shown in Figure 1(a), the same image corresponds to "Red", "Apple," and "Sweet" in different classification tasks. Multi-task learning is prevalently applied in various fields, such as computer vision (Meyerson & Miikkulainen, 2018; Chen et al., 2018), natural language processing (Liu et al., 2019; Collobert & Weston, 2008) and reinforcement learning (Hessel et al., 2019).


Task Annotation
+
Label Annotation
Task 1 (Color)
"Red" "Green" "Yellow" "Yellow" ...
Task 2 (Fruit)
"Apple" "Apple" "Lemon" "Banana" ...
Task 3 (Taste)
"Sweet" "Sour" "Sour" "Sweet" ...
Manual Assigned Data
Multi-Task Machine
"Sweet" "Green" "Yellow" "Banana" "Red" "Apple" "Sour" "Lemon" "Sour" "Apple"
Confusing Data
"Yellow" "Green" "Red" "Banana" "Lemon" "Apple" "Apple" "Sour" "Sour" "Sweet"
Data De-confuse
Task Understanding
+
Multi-Task Machine

(a) Traditional Multi-task Learning.

(b) Confusing Supervised Learning.

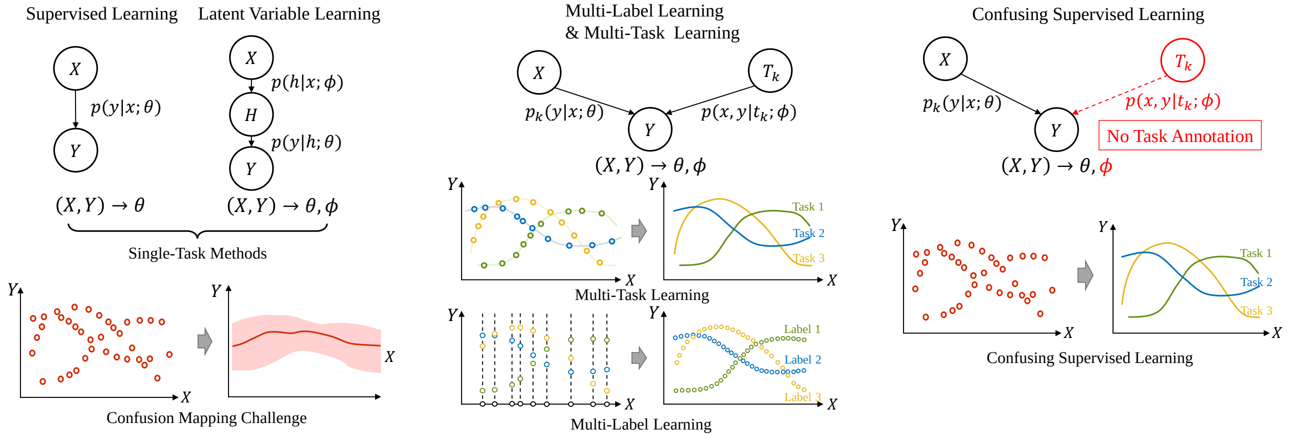
Figure 1. The learning paradigm of multi-task learning and confusing supervised learning.

Generally speaking, multi-task learning methods require manual definition of tasks and annotation for task encoding, as shown in Figure 1(a). When collecting a multi-task dataset, we need to construct the task definition and make the task annotations for every input-label sample. These manual task annotations require enormous annotation cost and constrain the generalization of multi-task machines. Moreover, humans also face confusingly labeled data in the real world, and they learn the high-level task understanding for better analysis and decision, which is a critical process in human recognition. Therefore, a novel, promising problem is whether the machine could understand task concepts from basic input-label pairs that contain neither task annotation nor the sample allocation for tasks, shown in Figure 1(b).

Without task annotation, training samples from multiple tasks are mixed together in a confusing manner, where the same inputs have different outputs. Then, traditional supervised learning fails to learn with this data due to a *confusing mapping challenge*. The traditional learning machine learns a function to approximate the unknown, and certain mapping from input to output minimizes the risk functional. However, the assumption of a single mapping function in

[1] Department of Automation, Tsinghua University, Beijing, 100084, China [2] LSBDPA Beijing Key Laboratory, Beijing, 100084. China [3] Beijing Innovation Center for Future Chip, Beijing, 100084, China. Correspondence to: Feng Chen <chenfeng@mail.tsinghua.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).



(a) Confusing Mapping Challenge. (b) Multi-task & Multi-label Learning. (c) Confusing Supervised Learning.

Figure 2. Comparison of traditional supervised learning, latent variable learning, multi-task learning, multi-label learning, and confusing supervised learning (CSL).

the basic theory (Vapnik, 2003) leads to unavoidable confusion risk for our multi-task confusing data. As a result, the machine could only learn the means of these multi-task outputs, instead of the exact mapping relation for any task. Our goal is to clarify these confusing supervised samples and complete task understanding.

In order to understand the task concept, we took confusing supervised data and proposed a novel learning method: *Confusing Supervised Learning (CSL)*. We found that proper allocation for confusing samples could prevent conflicting mapping relationships, which is consistent with the concept of tasks in human cognition. Following this idea, we constructed a CSL framework that contains two types of function variables: (i) Deconfusing Function and (ii) Mapping Functions. The deconfusing function represents the relationship between samples and tasks, which allocates confusing data into multiple tasks. The mapping functions represent the relationships from input to labels for each task. With these two function variables, the risk functional involves a reasonable upper-level sample allocation and accurate lower-level multi-task input-label mappings.

To achieve our goal, the CSL method must deal with two crucial difficulties: (i) whether it is feasible; and (ii) how to learn with it. For the first problem, we proved that in the CSL method, the expected risk functional minimization can be approximated by minimizing the empirical risk, and the optimal risk value could be reduced to zero. For the second problem, we constructed a *CSL-Net* for representing variables of CSL. However, the one-hot constraint of the outputs makes gradient back-propagation unfeasible. We transformed the CSL risk minimization into two non-increasing optimization problems. By alternatively performing training for these two optimizations, the iterative learning algorithm can get the solutions for the CSL-Net.

To verifty the advantages of our confusing supervised learning framework, we respectively constructed experiments for function regression and image recognition with obfuscated multi-task data, Our experimental results show that the CSL-Net can autonomously learn a human task concept and multiple mappings for every task simultaneously. Compared with multi-task learning with complete information, the CSL-Net could achieve the same complete cognition result from confusing data without task annotations.

2. Related Work

Multi-task Learning. Multi-task learning aims to learn multiple tasks simultaneously and improves learning efficiency and performance by sharing feature representations (Caruana, 1997; Argyriou et al., 2007; Evgeniou & Pontil, 2004; Long et al., 2017). Multi-task learning is prevalent in various fields including computer vision (Meyerson & Miikkulainen, 2018; Chen et al., 2018; Kendall et al., 2018), natural language processing(Hashimoto et al., 2017; Liu et al., 2019) and reinforcement learning(Hessel et al., 2019; Omidshafiei et al., 2017). In multi-task learning, the task to which every sample belongs is known, as shown in Figure 2(b). With this task definition, the input-output mapping of every task can be represented by a unified function. However, these task definitions are manually constructed, and machines need manual task annotations to learn. Without this annotation, our goal is to understand the task concept from confusing input-label pairs.

Considering learning from confusing data samples, latent variable learning and multi-label learning have some similarities but differ essentially in statistical theory.

Latent Variable Learning. The mapping relations in latent variable learning contain multiple distribution modules, and learning methods need to distinguish samples from different models (Kumar et al., 2010; Serban et al., 2017; Daumé III & Kumar, 2013). Latent variable learning focuses on mixed probability models, as shown in Figure 2(a). In essence, all input-label pairs come from a unified distribution, and this distribution is estimated by a mixture of multiple probability models. However, multi-task confusing samples are from different distributions. Due to the lack of task information, statistically speaking, the estimation of mapping parameters is insufficient for confusing samples shown in Figure 2(c).

Multi-label Learning. Multi-label learning considers situations where one object contains multiple semantic labels simultaneously(Boutell et al., 2004; Durand et al., 2019; Kazawa et al., 2005; Gopal & Yang, 2010). In this learning problem, one input variable is assigned a set of proper labels and the learning machine must judge which labels express its semantics correctly(Tsoumakas & Katakis, 2009). Different from our multi-task confusing data, multi-label learning assumes that each label judgment is an independent learning problem, which does not involve the semantic understanding of tasks in multi-task learning.

3. Confusing Supervised Learning

3.1. Confusing Mapping Challenge

Our goal is to understand the task concept with confusing multi-task data. However, traditional supervised learning fails to learn from confusing data due to confusing multiple mappings.

In the standard supervised learning problem, the learning goal is to select an optimal function from a set of functions to minimize the risk functional (Vapnik, 2003). Let the training samples be (x, y), which is from an identical but unknown mapping relationship $y = f(x)$ (or $p(y|x)$). Without loss of generality, let the risk measure for the samples be mean square error (MSE), that is $R_0 = (y - g(x))^2$. Then the expected risk functional is

$$R(g) = \int_x (f(x) - g(x))^2 p(x)\,\mathrm{d}x, \tag{1}$$

where $p(x)$ is the prior distribution of input variable x. Since the function $f(x)$ (or distribution $p(y|x)$) is unknown, the risk is estimated by data samples $(x_i, y_i), i = 1, ..., m$. Then supervised learning methods minimize the empirical risk

$$R_e(g) = \sum_{i=1}^{m} (y_i - g(x_i))^2 \tag{2}$$

to choose the optimal learning function for risk (1). Obviously, the theoretically optimal solution for risk functional is $g^*(x) = f(x)$, with which the minimum risk can be reduced to zero.

Now we consider the *Confusing Supervised Learning (CSL)* problem. The samples also appear in the form of $(x_i, y_i), i = 1, ..., m$, but they come from a number of different tasks $y = f_j(x), j = 1, ..., n$ (or $p_j(y|x), j = 1, ..., n$). These samples are mixed together and it is unknown which samples come from the same task f_j (or $p_j(y|x)$).

For such confusing data, the existing supervised learning methods face the theoretically unavoidable *confusion mapping challenge*, as shown in Figure 2(a). Specifically, the confusing samples can be expressed as $p(x, y) = P(f_j) \cdot p_j(y|x)p(x)$, where $P(f_j)$ is the prior probability of tasks f_j and $p_j(y|x)$ is the posterior probability of y with x in the task j. When using traditional supervised learning methods, the risk functional is

$$R(g) = \int_x \sum_{j=1}^{n} \underbrace{(f_j(x) - g(x))^2 p(f_j)}_{\text{Confusing Multiple Mappings}} p(x)\,\mathrm{d}x. \tag{3}$$

Then we calculate the theoretical extreme value of this risk functional. The optimal solution $g^*(x)$ is that

$$g^*(x) = \sum_{j=1}^{n} p(f_j) f_j(x) = \bar{f}(x). \tag{4}$$

This result means that the learned function is the mean of all ground-truth functions instead of every specific function which we need. Also, at this time, the minimum risk $R(g^*) > 0$, which means the existing supervised learning paradigm fails to learn the confusing supervised data and lead to an unavoidable confusion risk. How to overcome this confusion mapping challenge similar to human cognition is essential for confusing supervised learning.

3.2. Learning Functions and Risk Functional of CSL

In order to achieve the goal of CSL, we built a novel learning framework. We found that, in human cognition, confusing data are first allocated into different groups. After making a reasonable allocation, all samples in the same group could be represented by a unified mapping function without conflicting outputs for the same input. We think this allocation is the basic understanding of tasks that makes confusing risk reduced to zero.

Following this idea, we introduce two types of learning functions: (1) *Deconfusing Function* represents the allocation of which samples come from the same task; and (2) *Mapping Function* represents the mapping relation from input to output of every learned task.

Concretely, for the mapping function, a family of learning functions $\{g_k, k = 1, ..., l\}$ is introduced to represent multiple ground-truth mappings $\{f_j, j = 1, ..., n\}$ contained in confusing samples. The deconfusing function is defined as $h(x, y, g_k)$. This function is an indicator function to determine whether the sample (x, y) is assigned to the task g_k. With these two types of function variables, we modify the risk functional. As MSE loss is still used (other losses also can be applied), the risk functional of the CSL framework is defined as

$$R(g,h) = \int_x \sum_{j,k} (f_j(x) - \underbrace{g_k(x)}_{\substack{\text{Mapping}\\ \text{Function}}})^2 \underbrace{h(x, f_j(x), g_k)}_{\text{Deconfusing Function}} p(f_j)p(x)\,\mathrm{d}x. \tag{5}$$

As the probability term in risk functional is unknown, we instead estimate empirical risk with data samples that

$$R_e(g,h) = \sum_{i=1}^{m} \sum_{k=1}^{n} |y_i - g_k(x_i)|^2 \cdot h(x_i, y_i; g_k). \tag{6}$$

Compared with traditional supervised learning methods, the CSL framework has two differences. The first is that the mapping expression changes from one function to multiple functions. The second is the introduction of a deconfusing function. In risk measurement, the risk metric of every sample affects only the assigned learning task.

3.3. Discussion of Existence and Uniqueness

Intuitively, the expressive capability of the CSL framework is stronger than traditional supervised learning. Here we prove that this framework is sufficient to overcome the confusion risk in the CSL problem. We have the following theorem.

Theorem 1 (**Existence of Solution**). *With the confusing supervised learning framework, there is an optimal solution*

$$h^*(x, f_j(x), g_k) = I[j = k], \tag{7}$$

$$g_k^*(x) = f_k(x),\ k = 1, ..., n, \tag{8}$$

that makes the expected risk function of the CSL problem zero.

Proof. This is a constructive conclusion. We can obtain the result by directly taking a direct solution (7) and deconfusing function h into risk functional (5). □

Although this representation is sufficient for confusing data, there are some meaningless solutions in all optimal risk solutions. For instance, we exchange the mapping results from a local input $\hat{x}$ of two tasks in optimal solutions $g^*(\hat{x})$ that $g_i'(\hat{x}) = g_j^*(\hat{x})$ and $g_j'(\hat{x}) = g_i^*(\hat{x})$ while the mappings of other tasks and other inputs are kept. This is also an optimal, however meaningless, risk solution. Therefore, necessity constraints are needed to avoid meaningless trivial solutions. Fortunately, when implemented with neural networks (consisting of continuous operation modules), the mapping function contains a continuous tendency itself, which results in a set of meaningful solutions. In more complex cases, we can further add more necessity constraints for developing a better task understanding.

3.4. Determine the Number of Tasks

In our CSL framework, the form of learning functions differs with different task numbers. Since the task number for ground-truth is unknown, a crucial problem is the determination of the number of tasks when understanding task concepts using confusing supervised data. We refer to the analysis of generalization error in statistical learning theory and use the principle of structural risk minimization to determine the number of learned tasks in the CSL framework.

The following theorem demonstrates that the method of empirical risk minimization is valid in the CSL framework.

Theorem 2 (**Error Bound of CSL**). *With probability at least $1 - \eta$ simultaneously with finite VC dimension τ of CSL learning framework, the inequality*

$$R(\alpha) \le R_e(\alpha) + \frac{B\varepsilon(m)}{2}\left(1 + \sqrt{1 + \frac{4R_e(\alpha)}{B\varepsilon(m)}}\right) \tag{9}$$

holds true, where α is the total parameters of learning function g, h, B is the upper bound of one sample's risk, and

$$\varepsilon(m) = 4\frac{\tau(\ln\frac{2m}{\tau} + 1) - \ln\eta/4}{m}. \tag{10}$$

Proof. Note the samples as $z = (x, y)$, and note samples under the task j as $z^{(j)} = (x, f_j(x))$. The set of learning functions $g_k(x)$ and $h(z, g_k)$ are given by parametric form $\{g_k(x;\theta), \theta \in \Theta\}$ and $\{h(z, g_k; \phi), \phi \in \Phi\}$. Let the whole learning parameters α be $\alpha = (\theta, \phi)$. We mark the risk of one sample as

$$Q(z, k; \alpha) = (y - g_k(x;\theta))^2\, h(x, y, k; \phi). \tag{11}$$

Then we rewrite the expected risk functional that

$$R(g,h) = \int_z \Big[\sum_{j=1}^{n}\sum_{k=1}^{l} Q(z^{(j)}, k; \alpha) P(f_j)\Big] p(z^{(j)})\,\mathrm{d}z^{(j)}.$$

Also, the empirical functional is rewritten as

$$\begin{aligned} R_e(g,h) &= \sum_{i=1}^{m}\sum_{k=1}^{n} Q(z_i, k; \alpha) \\ &= \sum_{i=1}^{m}\sum_{k=1}^{n}\Big[\sum_{j=1}^{l} Q(z_i, k; \alpha | f_j) P(f_j)\Big] \\ &= \sum_{i=1}^{m}\Big[\sum_{k=1}^{n}\sum_{j=1}^{l} Q(z_i^{(j)}, k; \alpha) P(f_j)\Big]. \end{aligned}$$

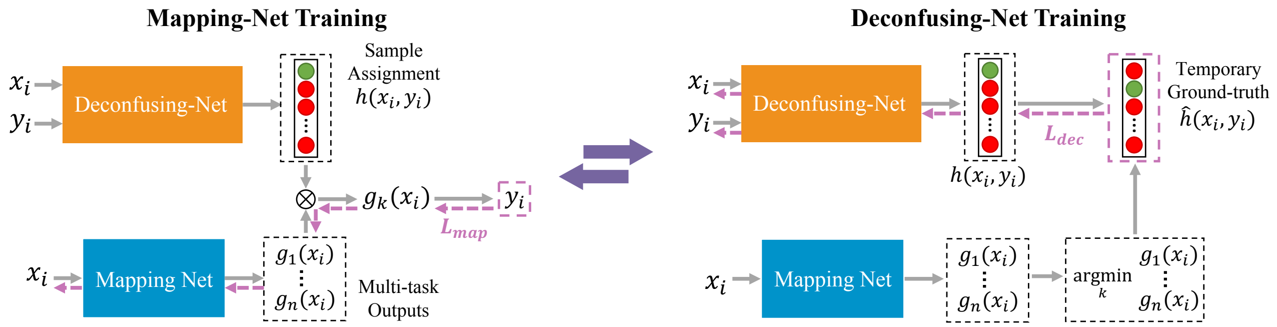


Figure 3. Training Process of the CSL-Net.

Mark that $\tilde{Q}(z^{(j)};\alpha) = \sum_{k=1}^{n}\sum_{j=1}^{l} Q(z^{(j)},k;\alpha)$. From the conclusion of the risk bound in statistical learning theory, we can get the result of the theorem. □

The above theorem proof explains that the risk functional minimization of CSL has the same form of error bounds as in statistical learning theory. Therefore, the structural risk minimization principle is also valid; that is, the larger the VC dimension of the learning functions are, the larger generalization error will be, even leading to overfitting of finite training samples. In the CSL framework, an important factor affecting the VC dimension is the assumed number of tasks. A small task number means a low VC dimension, which results in a high training risk and even fails to solve the confusion mapping challenge. On the other hand, a large number of tasks bring high VC dimensions, which leads to a small confidence interval. Therefore, the principle of determining the task is to choose the *minimum number* of tasks that makes the training risk as small as possible, which leads to the smallest guaranteed risk (9).

4. CSL-Net

In this section, we consider another crucial issue: how to implement and train a network for CSL.

4.1. The Structure of CSL-Net

This goal of the training algorithm is minimizing the empirical risk functional that

$$\min_{g,h} R_e = \sum_{i=1}^{m}\sum_{k=1}^{n}(y_i - g_k(x_i))^2 \cdot h(x_k, y_k; g_k). \qquad (12)$$

We used two neural networks, *deconfusing-net* and *mapping-net*, to implement two learning function variables in empirical risk. The mapping-net corresponded to functions set $g_k, k = 1, ..., n$. It is a multi-branch network and the output of every branch represents the mapping function $y_k = g_k(x)$ in one certain task. The deconfusing function h was implemented by deconfusing-net, whose input is a complete sample (x, y) and the output is an n-dimension one-hot vector. The output of deconfusing-net determined which task mapping g_k the input sample (x, y) should be assigned to. With these two nets, we could represent the risk functional, and the whole structure is named CSL-Net.

However, there is a core difficulty in that the risk functional with this structure cannot be optimized by gradient back-propagation. To ensure the physical meaning of deconfusing-net, the training of the deconfusing-net is under the constraint of a one-hot output. A one-hot output is a discontinuous function that cannot use gradient back-propagation. When using Softmax for approximation, the training makes the deconfusing-net output into a non-one-hot form. Such a combined output does not meet the meaning of sample assignment, resulting in meaningless trivial solutions. Therefore, we needed to design a novel training algorithm that can perform gradient back-propagation for two learning nets under the constraint of one-hot output.

4.2. Iterative Deconfusing Algorithm

For solving this training difficulty, we transformed the joint optimization problem of two networks into a pair of univariable optimization problems for two networks. In each single-network optimization step, we fixed the parameters of one network and updated the parameters of another. With one network's parameters unchanged, we rewrote the optimization problem as an equivalent form for another network such that the new problems can be solved by a gradient descent method of neural networks. We alternately performed the solving processes of this pair of equivalent optimization problems, thus progressively getting the solution of the original objective optimization (12). Now we showed the specific training algorithm of deconfusing-net and mapping-net.

Training of Mapping-Net. In the training of mapping-net, we maintained the parameters from deconfusing-net. When the function h is determined, the way of minimizing the original risk (12) is to train every mapping function g_k with the assigned samples (x_i^k, y_i^k), as shown on the left side of Figure 3. Then the optimization problem of mapping-net becomes the following:

$$\min_{g_k} L_{map}(g_k) = \sum_{i=1}^{m_k} |y_i^k - g_k(x_i^k)|^2, \quad k = 1, ..., n. \qquad (13)$$

This optimization problem can be solved by updating the mapping-net with a back-propagation algorithm.

Training of Deconfusing-Net. In the training of deconfusing-net, the parameters of the mapping-net were fixed. Since deconfusing-net outputs a one-hot vector, the original risk form (12) cannot be used to train directly. We found that, to minimize the original risk, every data sample (x, y) should be assigned to the function g_k where $g_k(x)$ is closest to y among all functions $g_k, k = 1, ..., n$. Therefore, during this training phase, mapping-net provided a temporary training object for deconfusing-net, that is,

$$\hat{h}(x_i, y_i) = \arg\min_k |y_i - g_k(x_i)|^2. \tag{14}$$

Then the deconfusing-net optimization became

$$\min_h L_{dec}(h) = \sum_{i=1}^{m} |h(x_i, y_i) - \hat{h}(x_i, y_i)|^2. \tag{15}$$

This process is shown on the right side of Figure 3. Obviously, this optimization problem can be solved by updating the subject-net with a back-propagation algorithm.

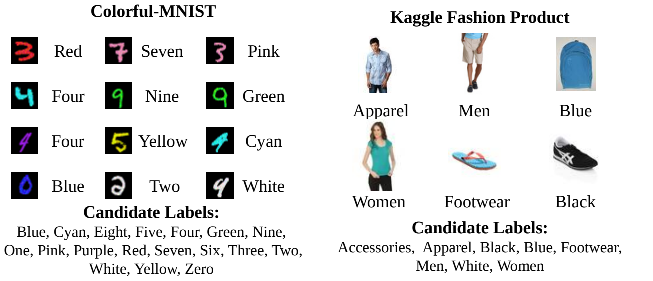
5. Experiment

5.1. Setup

We constructed a series of benchmarks for the CSL problem which includes function regression tasks and pattern recognition tasks.

Function Regression Tasks: For the traditional regression tasks, every input x corresponds to an output y, which are associated with a certain function $y = f(x)$. In the multi-task problem, there are multiple functions $\{f_j, j = 1, ..., n\}$. Every sample (x_i, y_i) is generated with one randomly selected mapping relationship $y = f_j(x)$. Then we received a set of confusing data samples $(x_i, y_i), i = 1, ..., m$, where every sample's corresponding task is unknown. The goal was to correctly determine task concept and sample allocation, as well as represent multi-task mapping function f_j.

Pattern Recognition Task: Pattern recognition tasks require the machine to learn classification capabilities, which predicts observed input x to the correct class label. In the CSL problem, there are multiple classification tasks for all inputs. Every observed sample only represents the classification result of one task, and which task the sample comes from is unknown. The goal is to understand the concept of multiple classification tasks from these confusing samples. Therefore, we built two datasets, *Colorful-Mnist* and *Kaggle Fashion Product* to evaluate learning methods in this CSL recognition problems.


Colorful-MNIST
Red
Seven
Pink
Four
Nine
Green
Four
Yellow
Cyan
Blue
Two
White
Candidate Labels:
Blue, Cyan, Eight, Five, Four, Green, Nine, One, Pink, Purple, Red, Seven, Six, Three, Two, White, Yellow, Zero
Kaggle Fashion Product
Apparel
Men
Blue
Women
Footwear
Black
Candidate Labels:
Accessories, Apparel, Black, Blue, Footwear, Men, White, Women

(a) Colorful MNIST Dataset (b) Kaggle Fashion Product Dataset

Figure 4. Data samples of two recognition experiments.

(1) *Colorful-Mnist*: We extended the MNIST dataset (LeCun et al., 1998) by adding random color in all images and obtained 0-9 digital images in 8 different colors, shown in Figure 4(a). Every confusing sample (x, y) contains one of the correct descriptions of input x from 18 candidate labels. From a human's perspective, ground-truth relations should be two criteria: color classification and number classification.

(2) *Kaggle Fashion Product*: We used a fashion dataset on Kaggle(Arslan et al., 2019) to construct a CSL recognition tasks for general objects, as shown in Figure 4(b). This dataset contains 9 basic candidate labels from 3 criteria. In human cognition, these labels could be divided into three criteria: gender, main category and main color.

5.2. Metrics of Confusing Supervised Learning

In order to quantitatively evaluate the performance of confusing supervised learning, we adopted human cognition as a performance ground-truth and built two metrics, which are *Task Prediction Accuracy* and *Label Predictions Accuracy*.

Task Prediction Accuracy. These metrics evaluate the task understanding from confusing data, by which we can infer the task conception from given samples. We let learned machines predict the learned task for confusing test samples and compared results to the human cognition ground-truth. Since the results of exchanged tasks' order are equivalent, we used the task prediction closest to humans' as the evaluation result, which is defined as:

$$\alpha_T(j) = \max_k \frac{1}{m} \sum_{i=1}^{m} I[h(x_i, y_i; f_k), \tilde{h}(x_i, y_i; f_j)]. \tag{16}$$

$\tilde{h}$ is the task cognition of humans. The higher task prediction accuracy means closer cognition to humans'.

Label Prediction Accuracy. Besides learning mapping allocation like humans, machines also need to accurately approximate every mapping function, so as to provide all corresponding labels. Therefore, we tested the prediction of test inputs under human high-level allocation rules. Every mapping contains its ground-truth output, and machines should predict the correct output close to the ground-truth. Considering the exchange equivalence, we define the fol-

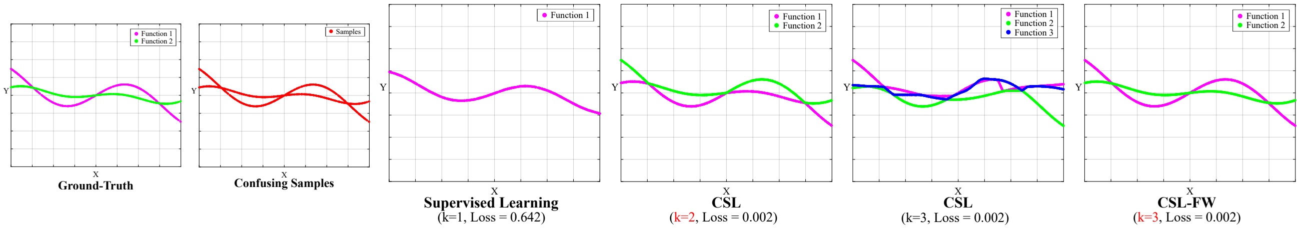


Figure 5. Results on two confusing functions.

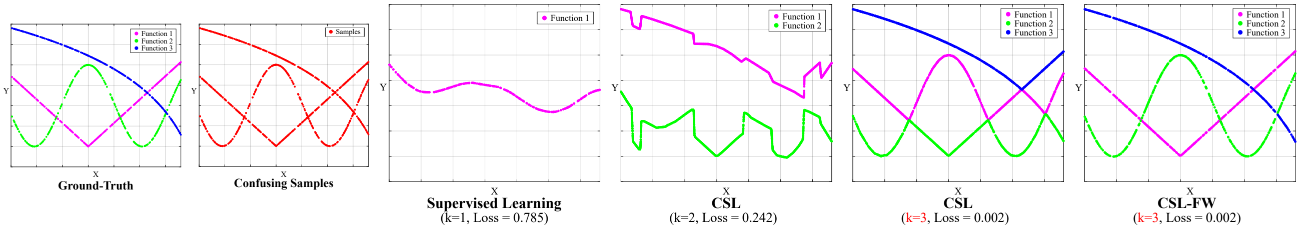


Figure 6. Results on three confusing functions.

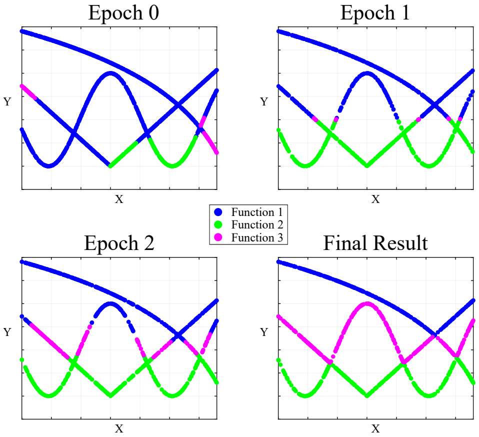


Figure 7. Training history of sample de-confusing.

lowing sample prediction accuracy for each mapping f_j that:

$$\alpha_L(j) = \max_k \frac{1}{m}\sum_{i=1}^{m} 1 - \frac{|g_k(x_i) - f_j(x_i)|}{|f_j(x_i)|}. \tag{17}$$

5.3. Results of Function Regression Tasks

In this experiment, we constructed the cases of two and three confusing functions and evaluated the performance of traditional supervised learning methods and CSL methods. The ground-truth multiple functions and confusing samples are shown on the left side of Figure 5 and Figure 6. The target of learning is to understand every function accurately. In this experiment, both mapping-net and deconfusing-net are made up of full-connect networks.

The results are shown on right side of Figure 5 and Figure6. Traditional supervised learning methods resulted in a mean value function when dealing with confusing samples, which verifies the confusion mapping challenge. On the other hand, the CSL method could divide these confusing samples into a reasonable function grouping and construct multiple continuous mapping functions. With the increase of learned task number k, the loss decreases and the suitable k^* corresponds to the smallest task number leading to zero learning risk, which verifies the conclusion from Section 3.4. Without extra constraints, although a normal CSL result is a reasonable continuous function solution, it differs from the ground-truth. This is the difficulty of the CSL problem, mentioned in Section 3.3, in that multiple solutions could lead the learning risk converging towards zero. Therefore, we introduced a few-shot (5-shot) warm-up to determine the initialization of the neural network. We found that the CSL methods with few-shot warm-up steadily learn the ground-truth results from confusing samples, shown in "CSL-FW" of Figure 5 and Figure 6. Additionally, the output result of deconfusing-net in the entire learning process is shown in Figure 7, which demonstrates deconfusing-net progressively understanding the task of three regressing functions.

5.4. Results of Pattern Recognition Tasks

In this experiment, we evaluated traditional supervised learning methods and CSL methods in the pattern recognition problem on Colorful-Mnist and Kaggle Fashion Product datasets. Two baselines are used as a comparison. Pseudo-Label(Lee, 2013) is a semi-supervised learning method by assuming the labels of unsupervised samples, and SMiLE(Tan et al., 2017) is a generalized form of label propagation algorithm. Table 1 shows the results of various learning methods with both confusing data and task annotated data.

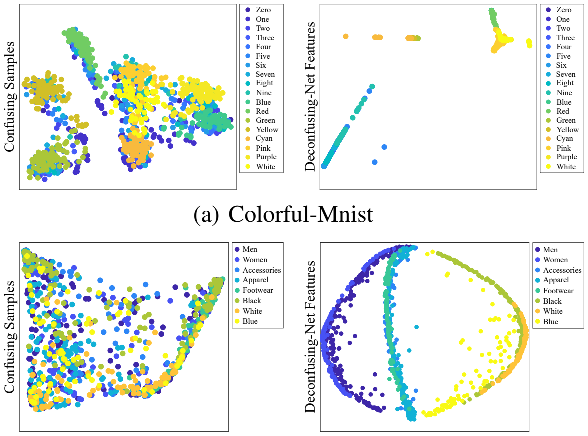
Colorful-MNIST. From the results of learning with confusing data, only the CSL method understands these two tasks, and further learns an accurate classification capability. In contrast, without task understanding the traditional supervised learning methods (Trad SL) and other learning methods learn confusing results that an input only corre-

Table 1. Accuracy of Pattern Recognition Experiments.

Learning Methods		Colorful-MNIST				Kaggle Fashion Product					
		$\alpha_T(1)$ (Cor)	$\alpha_T(2)$ (Num)	$\alpha_L(1)$ (Cor)	$\alpha_L(2)$ (Num)	$\alpha_T(1)$ (Gen)	$\alpha_T(2)$ (Cat)	$\alpha_T(3)$ (Cor)	$\alpha_L(1)$ (Gen)	$\alpha_L(2)$ (Cat)	$\alpha_L(3)$ (Cor)
Confusing Data	Trad SL	/	/	39.25	52.50	/	/	/	23.59	42.64	29.17
	Pseudo-Label	/	/	36.57	50.01	/	/	/	20.74	33.41	26.30
	SMiLE	/	/	12.94	19.98	/	/	/	16.04	32.74	18.41
	CSL	**98.24**	**99.02**	**99.32**	**97.18**	**98.42**	**99.16**	**98.90**	**93.25**	**97.87**	**90.84**
Task Annotated	Trad MT	99.48	99.61	99.24	98.15	99.01	99.43	99.17	92.91	97.82	91.64
	ML-LOC	99.57	99.58	99.66	98.62	99.12	98.92	99.25	94.54	98.63	94.12

Table 2. Accuracy of Partial Labeled Multi-label Learning.

Methods	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	Macro-Ave
Pseudo-Label	81.80	77.61	90.85	76.87	78.15	82.24	75.47	71.86	79.36
SMiLE	59.10	76.87	48.18	76.70	71.84	63.22	71.24	62.86	66.25
CSL	**98.18**	**98.25**	**94.92**	**95.28**	**99.10**	**93.27**	**94.32**	**92.75**	**95.76**
All-Label	99.68	97.98	94.65	94.86	98.82	93.45	93.74	93.26	95.80



(a) Colorful-Mnist

(b) Kaggle Fashion Product

Figure 8. Spectral embedding results of confusing samples and deconfusing-net's features in pattern recognition experiments.

sponds to one label in ”Color” pattern or ”Number” pattern, leading to low accuracy on evaluation metrics. When further comparing the CSL results to that of multi-task learning with task annotation, we find that the CSL method learns almost the same task understanding and classification results just from confusing data.

Kaggle Fashion Product. This experiment is in the same form of Colorful-MNIST, but the number of subjects increased and images became more complicated and more practical. In this experiment, we used the pre-trained CNN backbone and trained the full-connect networks following the CNN features. Beyond the confusing results of traditional learning methods, the CSL methods autonomously learned three tasks which exactly correspond to “Gender”, “Category”, and “Color” in human cognition. This experiment demonstrates the task understanding capability of the CSL method in practical multi-task recognition problems.

We further visualized the confusing samples and learned features from deconfusing-net by spectral embedding, shown in Figure 8. The result demonstrates that the CSL-Net can reasonably separate the original confusing samples through task understanding.

5.5. Application of Multi-label Learning

Besides the amazing capability of task understanding without task annotation, the CSL method also has advantages in traditional learning problems such as partially labeled multi-label learning. In multi-label classification, since multi-label annotation is difficult, an alternative strategy is learning multi-label classification from partial labels (Deng et al., 2014). In the Kaggle Fashion Product experiment, every image contained multiple correct labels, while every sample only gave one of them (partially labeled). Therefore, we evaluated the result of CSL methods on multi-label metrics. As shown in Table 2, by learning human-like cognition, CSL methods outperform other multi-label learning methods on Macro-Average accuracy.

5.6. Limitations

The Number of Tasks. As analyzed in Section 3.4, we determined the task number by increasing the assumed number of tasks progressively, and the lowest task number that brings the risk closest to zero is the best. However, this method requires repeated training processes. A promising idea is adding low-quality constraints for deconfusing-net to ensure that the optimal risk is obtained with the smallest number of tasks.

Learning of Basic Features. In our experiment of pattern recognition, we only trained the full-connect network based on learned CNN features. We found that the current algorithm is difficult for learning basic features directly through a CNN structure and understand tasks simultaneously. However, this difficulty does not affect the effectiveness of our algorithm in learning confusing data based on pre-trained features, and it is an open question for future work.

6. Conclusion

We proposed a novel learning idea to understanding tasks from basic input-label pairs without manual task annotations. Following the characteristics of human cognition, the machine learned the minimum risk for confusing samples by differentiating multiple mappings, thereby obtaining basic task concepts. We believe that the amazing result in this paper is an important advantage for achieving artificial general intelligence and the CSL method will be applied in more machine learning problems in the future.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61671266, Grant 61836004, and in part by the Institute Guo Qiang, Tsinghua University under Grant 2019GQG0006.

References

Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. In *Advances in neural information processing systems*, pp. 41–48, 2007.

Arslan, H. S., Sirts, K., Fishel, M., and Anbarjafari, G. Multimodal sequential fashion attribute prediction. *Information*, 10(10):308, 2019.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pp. 793–802, 2018.

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Daumé III, A. K. H. and Kumar, A. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning*, pp. 1723–1730, 2013.

Deng, J., Russakovsky, O., Krause, J., Bernstein, M. S., Berg, A., and Fei-Fei, L. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3099–3102. ACM, 2014.

Durand, T., Mehrasa, N., and Mori, G. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 647–657, 2019.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117. ACM, 2004.

Gopal, S. and Yang, Y. Multilabel classification with metalevel features. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 315–322. ACM, 2010.

Hashimoto, K., Tsuruoka, Y., and Socher, R. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1923–1933, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hessel, M., Soyer, H., Espeholt, L., Czarnecki, W., Schmitt, S., and van Hasselt, H. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019.

Kazawa, H., Izumitani, T., Taira, H., and Maeda, E. Maximal margin labeling for multi-topic text categorization. In *Advances in neural information processing systems*, pp. 649–656, 2005.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pp. 1189–1197, 2010.

Kurzweil, R. The singularity is near: When humans transcend biology. *Cryonics*, 85(1):160–160, 2005.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 2–8, 2013.

Liu, P., Chang, S., Huang, X., Tang, J., and Cheung, J. C. K. Contextualized non-local neural networks for sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6762–6769, 2019.

Long, M., Cao, Z., Wang, J., and Philip, S. Y. Learning multiple tasks with multilinear relationship networks. In *Advances in neural information processing systems*, pp. 1594–1603, 2017.

Meyerson, E. and Miikkulainen, R. Pseudo-task augmentation: From deep multitask learning to intratask sharing-and back. In *International Conference on Machine Learning*, pp. 3508–3517, 2018.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2681–2690. JMLR. org, 2017.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3295–3301, 2017.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., and Lanctot, M. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Tan, Q., Yu, Y., Yu, G., and Wang, J. Semi-supervised multi-label classification using incomplete label information. *Neurocomputing*, 260:192–202, 2017.

Tsoumakas, G. and Katakis, I. Multi-label classification. In *Database Technologies: Concepts, Methodologies, Tools, and Applications (4 Volumes)*, pp. 309–319, 2009.

Vapnik, V. N. Statistical learning theory. *Annals of the Institute of Statistical Mathematics*, 55(2):371–389, 2003.