Hacking Task Confounder in Meta-Learning

Jingyao Wang\textsuperscript{1,2}, Yi Ren\textsuperscript{1}, Zeen Song\textsuperscript{1,2}, Jianqi Zhang\textsuperscript{1,2}, Changwen Zheng\textsuperscript{1} and Wenwen Qiang\textsuperscript{1,2}\textsuperscript{*}

\textsuperscript{1}Institute of Software Chinese Academy of Sciences
\textsuperscript{2}University of Chinese Academy of Sciences

\{wangjingyao23, renyi, songzeen, zhangjianqi, changwen, qiangwenwen\}@iscas.ac.cn

Abstract

Meta-learning enables rapid generalization to new tasks by learning knowledge from various tasks. It is intuitively assumed that as the training progresses, a model will acquire richer knowledge, leading to better generalization performance. However, our experiments reveal an unexpected result: there is negative knowledge transfer between tasks, affecting generalization performance. To explain this phenomenon, we conduct Structural Causal Models (SCMs) for causal analysis. Our investigation uncovers the presence of spurious correlations between task-specific causal factors and labels in meta-learning. Furthermore, the confounding factors differ across different batches. We refer to these confounding factors as “Task Confounders”. Based on these findings, we propose a plug-and-play Meta-learning Causal Representation Learner (MetaCRL) to eliminate task confounders. It encodes decoupled generating factors from multiple tasks and utilizes an invariant-based bi-level optimization mechanism to ensure their causality for meta-learning. Extensive experiments on various benchmark datasets demonstrate that our work achieves state-of-the-art (SOTA) performance. The code is provided in https://github.com/WangJingyao07/MetaCRL.

1 Introduction

Meta-learning aims to develop models that can be rapidly transferred to previously unseen tasks. To achieve this, it first learns from diverse tasks to obtain models with high learning capacities. Then, it fine-tunes these models with little data from unseen tasks to obtain the desired ones. Recently, meta-learning has been widely applied in various fields, e.g., affective computing [Li et al., 2023], image classification [Qiang et al., 2021; Wang et al., 2023], and robotics [Schrum et al., 2022].

During the training phase, each batch consists of a series of randomly sampled $N$-way $K$-shot tasks, where $N$ denotes the number of classes per task and $K$ denotes the number of samples per class. The samples in each task are divided into a support set and a query set. Then, meta-learning models are trained in a bi-level optimization manner [Wang et al., 2021; Wang et al., 2023]. In brief, at the first level, the desired model for each task is fine-tuned by training on the support set using the meta-learning model. At the second level, the meta-learning model is learned using the query sets from all training tasks and the corresponding expected models for each task. Therefore, a widely adopted hypothesis is that as training progresses, the meta-learning model will acquire richer knowledge that can be transferred well to downstream tasks, achieving better performance [Rivoli et al., 2022].

However, our toy experiments reveal a conflicting phenomenon, i.e., the knowledge learned from the training tasks may be harmful to the unseen test tasks (See Subsection 3.2 for more details). Specifically, we first randomly sample 400 tasks from miniImageNet dataset [Vinyals et al., 2016] and divide them into a training set and a test set. Then, we define a metric $R_{i,j}$ to evaluate whether the meta-learning model trained on the training tasks can perform better on the test task, i.e., quantify the knowledge transfer performance from the training tasks to each test task. If $R_{i,j} < 1$, the learned knowledge from the training task can help improve the model performance on the test task (positive knowledge transfer), while $R_{i,j} > 1$ implies the learned knowledge is harmful to the test task (negative knowledge transfer). We...
Meta-learning aims to learn general knowledge from various training tasks, and then generalize to new tasks based on the acquired knowledge. Typical methods can be categorized into two types: optimization-based [Finn et al., 2017; Nichol and Schulman, 2018; Guo et al., 2024] and metric-based [Snell et al., 2017; Sung et al., 2018; Chen et al., 2020] methods. They both rely on shared structures and level learning mechanisms to learn general knowledge, resulting in remarkable performance on new tasks. However, meta-learning still faces the crisis of performance degradation. Various approaches have been proposed to address this issue, such as adding adaptive noise [Lee et al., 2020], reducing inter-task disparities [Jamal and Qi, 2019], limiting the trainable parameters [Yin et al., 2019; Oh et al., 2020], and task augmentation [Yao et al., 2021]. Despite alleviating performance degradation, they ignore the interaction between tasks, which is shown to be crucial in Section 3. In this study, we analyze the knowledge transfer effects between different training tasks with causal theory, and focus on the fundamental causes of performance degradation in meta-learning.

Causal learning aims to explore the causal relationships between variables in machine learning, modeling the target with a directed acyclic graph, also known as a causal model. It has been shown to aid models in unearthing underlying causal factors [Yang et al., 2021; Zhang et al., 2020; Nogueira et al., 2022]. Recent research attempts to combine causal knowledge with meta-learning methods to address domain challenges. Yue et al. [Yue et al., 2020] removed performance limitations of pre-trained knowledge through backdoor regulation. Ton et al. [Ton et al., 2021] utilized causal knowledge to distinguish causes and effects in a bivariate environment with limited data. Jiang et al. [Jiang et al., 2022] used causal graphs to remove undesirable memory effects. While they all combine meta-learning and causal learning, their focus is on addressing problems that differ from ours.

2 Related Work

Meta-learning aims to learn general knowledge from various training tasks, and then generalize to new tasks based on the acquired knowledge. Typical methods can be categorized into two types: optimization-based [Finn et al., 2017; Nichol and Schulman, 2018; Guo et al., 2024] and metric-based [Snell et al., 2017; Sung et al., 2018; Chen et al., 2020] methods. They both rely on shared structures and level learning mechanisms to learn general knowledge, resulting in remarkable performance on new tasks. However, meta-learning still faces the crisis of performance degradation. Various approaches have been proposed to address this issue, such as adding adaptive noise [Lee et al., 2020], reducing inter-task disparities [Jamal and Qi, 2019], limiting the trainable parameters [Yin et al., 2019; Oh et al., 2020], and task augmentation [Yao et al., 2021]. Despite alleviating performance degradation, they ignore the interaction between tasks, which is shown to be crucial in Section 3. In this study, we analyze the knowledge transfer effects between different training tasks with causal theory, and focus on the fundamental causes of performance degradation in meta-learning.

Causal learning aims to explore the causal relationships between variables in machine learning, modeling the target with a directed acyclic graph, also known as a causal model. It has been shown to aid models in unearthing underlying causal factors [Yang et al., 2021; Zhang et al., 2020; Nogueira et al., 2022]. Current research attempts to combine causal knowledge with meta-learning methods to address domain challenges. Yue et al. [Yue et al., 2020] removed performance limitations of pre-trained knowledge through backdoor regulation. Ton et al. [Ton et al., 2021] utilized causal knowledge to distinguish causes and effects in a bivariate environment with limited data. Jiang et al. [Jiang et al., 2022] used causal graphs to remove undesirable memory effects. While they all combine meta-learning and causal learning, their focus is on addressing problems that differ from ours.

3 Problem Formulation and Analysis

In this section, we first present the notation and problem definition of meta-learning. Next, we conduct experiments to evaluate the interaction between different tasks and illustrate the empirical evidence, i.e., the knowledge learned from the training tasks may be harmful to the unseen test tasks, reducing generalization performance. Finally, we construct SCMs to explore the reasons behind the empirical evidence.

3.1 Preliminaries

Given a task distribution \( p(T) \), the meta-training dataset \( D_{tr} \), and the meta-test dataset \( D_{ts} \), are all sampled from \( p(T) \) without class-level overlap. During the training phase of ML,
each batch contains $N_{tr}$ tasks, denoted as $\{\tau_i\}_{i=1}^{N_{tr}} \in \mathcal{D}_{tr}$,
and each task $\tau_i$ consists of a support set $\mathcal{D}_{i}^s = (X_i^s, Y_i^s) = \{(x_{i,j}^s, y_{i,j}^s)\}_{j=1}^{N_i^s}$
and a query set $\mathcal{D}_{i}^q = (X_i^q, Y_i^q) = \{(x_{i,j}^q, y_{i,j}^q)\}_{j=1}^{N_i^q}$, where $(x_{i,j}, y_{i,j})$ represents the sample and the corresponding label, and $N_i$ denotes the number of the samples. The meta-learning model $f_\theta = h \circ g$ utilizes the feature encoder $g$ and the classifier $h$ to learn the above tasks.

The learning mechanism of meta-learning is regarded as a bi-level optimization process. At the first level, it fine-tunes the desired model $f_{\theta, t}$ for task $\tau_i$ by training on the support set $\mathcal{D}_{i}^s$ using the meta-learning model $f_\theta$, presented as:

$$f_{\theta, t} \leftarrow f_\theta - \alpha \nabla_{f_\theta} \mathcal{L}(Y_i^s, X_i^s, f_\theta)$$

where $\alpha$ is the learning rate. At the second level, the meta-learning model $f_\theta$ is learned using the query sets $\mathcal{D}_{i}^q$ from all training tasks and the expected models for each task:

$$f_\theta \leftarrow f_\theta - \beta \nabla_{f_\theta} \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y_i^q, X_i^q, f_{\theta, t})$$

where $\beta$ is the learning rate. Note that $f_{\theta, t}$ is obtained by taking the derivative of $f_\theta$, so $f_{\theta, t}$ can be regarded as a function of $f_\theta$. Therefore, the update of $f_\theta$ mentioned in Eq.2 can be viewed as calculating the second derivative of $f_\theta$.

### 3.2 Empirical Evidence

From above and [Wang et al., 2021], meta-training on one batch can be viewed as a multi-task learning process. Meanwhile, a well-learned model should contain knowledge of all training tasks. Therefore, intuitively, one might assume that as training progresses, the meta-learning model will acquire richer knowledge (related to all tasks) and transfer better to downstream tasks, achieving great generalization. However, our toy experiments reveal that this is not always true.

Before introducing the toy experiments, we first present a method to quantitatively measure the knowledge transferred from one task to another. For task $\tau_i$, the model $f_\theta$ uses the support set $\mathcal{D}_{i}^s$ to obtain $f_{\theta, t}$ via Eq.1. Here, $f_{\theta, t}$ is considered to integrate the knowledge of task $\tau_i$ into $f_\theta$. Then, for task $\tau_j$, we first obtain the model $f_{\theta, t}^{1,2}$ by training $f_{\theta, t}$ on the support set $\mathcal{D}_{i}^s$, and then obtain the model $f_{\theta, t}^{2,2}$ by training $f_{\theta, t}$ on $\mathcal{D}_{j}^q$. Next, we calculate their losses on the query set $\mathcal{D}_{i}^q$, expressed as $\mathcal{L}(\mathcal{D}_{i}^q, f_{\theta, t}^{1,2})$ and $\mathcal{L}(\mathcal{D}_{i}^q, f_{\theta, t}^{2,2})$, respectively. Finally, we can estimate the ratio between these two losses, denoted as $R_{i,j}$, which quantifies the performance of knowledge transfer from task $\tau_i$ to task $\tau_j$. Thus, we have:

$$R_{i,j} = \frac{\mathcal{L}(\mathcal{D}_{j}^q, f_{\theta, t}^{1,2})}{\mathcal{L}(\mathcal{D}_{j}^q, f_{\theta, t}^{2,2})}$$

if $R_{i,j} < 1$, it means that task $\tau_i$ has a positive knowledge transfer effect on task $\tau_j$. On the other hand, if $R_{i,j} > 1$, it indicates the negative knowledge transfer effect of $\tau_i$ on $\tau_j$.

Next, we conduct experiments based on the quantitative method described above. We first randomly sample 400 tasks from minImageNet dataset, which are divided into a training set of 300 tasks and a test set of 100 tasks. Then, we use MAML as the baseline to calculate the score of $R_{i,j}$ from the training tasks to each test task in the middle of training.

Figure 1 shows the histograms of the knowledge transfer in the training phase of meta-learning along with exemplary tasks. From the results, we observe that as training proceeds, although the knowledge transfer effects become more and more positive, there always exists negative knowledge transfer between different tasks. It indicates that the training process of meta-learning cannot always obtain effective knowledge for unseen test tasks, and the aforementioned intuitive hypothesis is limited. Note that we also conduct experiments under various different settings, including using multiple meta-learning baselines, using different datasets, and training on multiple tasks simultaneously (the effect of multiple training tasks to a single test task), the impact of negative knowledge transfer always exists. More details and the full results are provided in Appendix F.

### 3.3 Causal Analysis and Motivation

To explore the reasons behind the above phenomenon, we propose using causal theory for analysis. We first construct a Structural Causal Model (SCM) based on the ground-truth causal mechanisms [Suter et al., 2019; Hu et al., 2022], as shown in Figure 2a. Specifically, this SCM contains two tasks $\tau_i$ and $\tau_j$, where $Y_i$ and $Y_j$ denote the label variables for tasks $\tau_i$ and $\tau_j$, $X_i$ and $X_j$ signify the corresponding generated samples for these two tasks, respectively. Meanwhile, $A^i$ and $A^j$ represent the distinct sets of causal factors specific to tasks $\tau_i$ and $\tau_j$, while $B^{i,j}$ encompasses shared causal factors. In this SCM, we assume that the samples $X_i$ and $X_j$ are generated by disentangled causal mechanisms using the causal factors, then $p(X_i|A^i, B^{i,j}) = \prod_k p(X_i|A^i_k) \prod_{t} p(X_i|B^{i,j}_t)$, where $A^i_k$ denotes the $k$-th factor of $A^i$, and $B^{i,j}_t$ denotes the $t$-th factor of $B^{i,j}$. Since $A^i$, $A^j$, and $B^{i,j}$ represent high-level knowledge of the data, we could naturally define the task label variable $Y_i$ for task $i$ as the cause of the $B^{i,j}$ and $A^i$. For the task $\tau_i$, we call $B^{i,j}$ and $A^i$ as the causal feature variables that are causally related to $Y_i$, and we call $A^j$ as the non-causal feature variables to task $\tau_i$. Therefore, we have $p(X_i|A^i, B^{i,j}, A^j) = p(X_i|A^i, B^{i,j})$.

Based on the proposed SCM, an ideal meta-learning predictor for each task should only utilize causal factors and be invariant to any intervention on non-causal factors. However, the joint learning of multiple tasks in meta-learning could give rise to the issue of using non-causal factors for unseen tasks, also known as spurious correlations, thereby making it challenging to achieve optimal predictions. To verify this claim, we consider the scenario of two binary classification tasks for simple but clear explanations. Let $Y_i$ and $Y_j$ be variables from $\{+1\}$, we assume $\tau_i$ and $\tau_j$ have non-overlapping factors, i.e., $B^{i,j} = \emptyset$, and the elements in $A^i$ and $A^j$ satisfy the constraint of Gaussian distribution. Then, we have:

**Theorem 1.** If the correlation between $Y_i$ and $Y_j$ is not equal to 0.5, the optimal classifier has non-zero weights for non-causal factors for each task. If the correlation between $Y_i$ and $Y_j$ equals 0.5 with limited training data, the optimal classifier
also has non-zero weights for non-causal factors in each task.

As inferred from the aforementioned theorem, the learned model leverages the causal factors from other tasks to facilitate the learning of the target task. Taking the task \( \tau_i \) as an example, the meta-learning model uses the causal factors \( \Lambda^i \) belonging to the task \( \tau_j \) for learning \( Y_j \). Therefore, there is a spurious correlation between \( \Lambda^i \) and \( Y_i \), which can be represented as a spurious path \( \Lambda^i \rightarrow Y_i \). Similarly, we can obtain the spurious path \( \Lambda^i \rightarrow Y_j \) for task \( \tau_j \). These spurious correlations are called “task confounders”, which are the reasons that lead to negative knowledge transfer in Subsection 3.2. The learning process can be viewed as the inverse process of the generating mechanism. Therefore, we can obtain the SCM with two spurious paths as illustrated in Figure 2b, which reflects the internal mechanism of task confounders in multi-task learning. The proof is provided in Appendix A.

4 Methodology

Based on the above analysis, we know that task confounders cause spurious correlations between causal factors and labels. An ideal meta-learning model should identify knowledge that is causally related to each task and learn from the identified multi-task knowledge. Therefore, we propose MetaCRL, a plug-and-play meta-learning causal representation learner that can encode decoupled causal factors for more efficient ML. It consists of two modules: (i) the disentangling module which aims to extract generating factors and eliminate task confounders; and (ii) the causal module which aims to ensure the causality of the obtained generating factors. In this section, we first introduce the disentangling module and the causal module in Subsections 4.1 and 4.2, respectively. Next, we provide the overall objective in Subsection 4.3. The pseudocode and pipeline of MetaCRL are shown in Appendix B.

4.1 Disentangling Module

In this module, we aim to obtain the whole generating factors related to all tasks and the task-specific generating factors related to each single task. Specifically, we first obtain the whole generating factors by learning a semantic matrix \( \Xi \). Next, we use a grouping function \( f_{gr} \) to acquire subsets of generating factors relevant to every single task. Note that this module does not guarantee the causality of the obtained generating factors, which will be addressed in the causal module.

For a pre-trained encoder, different channels of the feature representations are related to different kinds of semantics [Islam et al., 2020]. Thus, we propose to use the feature representation to learn the generating factors. During the training phase, we denote the \( N_{tr} \) training tasks as \( \{ \tau_i \}_{i=1}^{N_{tr}} \). Suppose that the number of generating factors is \( N_\xi \), then, we propose obtaining these \( N_\xi \) factors through the learning of a matrix \( \Xi \in \mathbb{R}^{N_\xi \times N_k} \). Here, \( N_\xi \) represents the dimension of the feature representation, i.e., the output dimension of the encoder \( g \), and each column of \( \Xi \) represents a distinct factor. Based on \( \Xi \), we can obtain a new representation of each sample, which can be called a generating representation, e.g., the generating representation for \( x_{i,j} \) can be presented as \( \Xi \cdot g(x_{i,j}) \).

Generally, generating factors in geometric space can be conceptualized as coordinate basis vectors, where each generating factor corresponds to a specific basis vector [Jensen and Shen, 2004]. Moreover, different coordinate bases can undergo mutual transformations via a reversible matrix, implying their equivalence. Hence, learning a task-specific matrix, serving as a base matrix, allows us to approximate task-related generating factors. Therefore, for \( \Xi \) to be considered a generating factor matrix, we need to constrain the column vectors of \( \Xi \) to be orthogonal to each other. Then we have:

\[
\mathcal{L}_{DM}(\Xi) = \sum_{i=1}^{N_{tr}} \sum_{j=i+1}^{N_{tr}} \Xi_{i,j}^T \Xi_{i,j} \tag{4}
\]

where \( \Xi_{i,j} \) represents the \( i,j \)-th column of \( \Xi \). Minimizing \( \mathcal{L}_{DM}(\Xi) \) makes the different columns of \( \Xi \) orthogonal to each other, thus leading \( \Xi \) to be task-related generating factors.

Next, for all the \( N_{tr} \) training tasks, the generating factors should be divided into \( N_{tr} \) overlapping groups, and each group corresponds to a task. To obtain these groups, we propose a learnable grouping function \( f_{gr} \), which is implemented using Multi-Layer Perceptrons (MLPs) to acquire task-specific generating factors. Take task \( \tau_i \) as an example, we first calculate the average sample \( x_i \) for this task, i.e.,

\[
x_i = \frac{1}{N_i+N_j} (\sum_{j=1}^{N_\xi} x_{i,j}^s + \sum_{j=1}^{N_\xi} x_{i,j}^q) \tag{5}
\]

Then, we input \( x_i \) into the encoder \( g, \Xi \), and \( f_{gr}(\Xi^T g(x_i)) \), yielding a vector with all elements greater than zero and matching the dimensionality of the generating representation. Then, each element is subject to the normalization operation, denoted as \( \text{Norm}(\cdot) \). As a result, the individual elements of the output vector, i.e., \( \text{Norm}(f_{gr}) \), can be interpreted as the probabilities that each generating factor belongs to task \( \tau_i \).

Note that each task is associated with a subset of factors in \( \Xi \) and can vary significantly from task to task. Meanwhile, the above calculation process of \( \Xi \) and \( f_{gr} \) may lead to degenerate solutions, e.g., the subset of generating factors for each task is the same. To address this issue, we propose a regularization term that consists of a \( L_1 \) norm and an entropy term, constraining the output of \( f_{gr} \) to be sparse and diverse. By minimizing the \( L_1 \) norm, we make the output of \( f_{gr} \) sparse, ensuring that each subset of generating factors are relevant to each single task. By maximizing the entropy term, we make the output of \( f_{gr} \) diverse, preventing the acquisition of task-specific generating factors suffering from degenerate solutions. The regularization term is:

\[
\mathcal{L}_{DM}(f_{gr}) = \sum_{i=1}^{N_{tr}} \left( \| f_{gr}(\Xi^T g(x_i)) \|_1 - \text{Entropy}(\sum_{j=1}^{N_\xi} f_{gr}(\Xi^T g(x_i))) \right) \tag{5}
\]

where \( f_{gr}(\Xi^T g(x_i)) \) represents the \( j \)-th element of the output of \( f_{gr} \). Through Eq.5, we obtain accurate task-specific generating factors, thus eliminating task confounders.

By combining Eq.4 and Eq.5, we obtain the loss of the disentangling module which can be expressed as:

\[
\mathcal{L}_{DM}(f_{gr}, \Xi) = \lambda_1 \cdot \mathcal{L}_{DM}(\Xi) + \lambda_2 \cdot \mathcal{L}_{DM}(f_{gr}) \tag{6}
\]

where \( \lambda_1 \) and \( \lambda_2 \) denote the loss weights of \( \mathcal{L}_{DM}(\Xi) \) and \( \mathcal{L}_{DM}(f_{gr}) \), respectively. Through the above process with three constraints, i.e., correlation, sparsity, and diversity, we can accurately obtain all the generating factors and the task-specific generating factors without task confounders.
4.2 Causal Module

In this module, we aim to ensure the causality of the generating factors obtained in the disentangling module. Following [Koyama and Yamaguchi, 2020], a model invariant to different distributions can learn causal correlations. Meanwhile, based on Theorem 9 described in [Arjovsky et al., 2019], by enforcing invariance over multiple training datasets that exhibit distribution shifts, the task-specific models could only use task-related causal factors and assign zero weights to those non-causal generating factors. Therefore, the causal module is designed to facilitate causal learning by using this invariance, thereby ensuring the causality of the generating factors obtained by $\Xi$ and $f_{gr}$.

During the training phase of ML, the training data can be divided into multiple support sets and query sets. As they comprise different samples, they can be regarded as different data distributions with distributional shifts. Meanwhile, the learning process of meta-learning can be depicted as follows: First, for every $f_0$, optimizing Eq.1 can achieve an optimal $f_{h_0}$ and $\mathcal{L}(Y^s, X^s, f_{h_0})$ on the support set. Next, altering the value of $f_0$ impacts the optimal $f_{h_0}$, we seek the optimal $f_0$ to obtain the optimal $f_{h_0}^*$ by optimizing $\frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y^q, X^q, f_{h_0})$ on the query sets (Eq.2). Thus, the bi-level optimization of Eq.1 and Eq.2 can be interpreted as achieving optimality across multiple datasets using the same $f_0$, and the causal factors are invariant on the support and query sets of the same task.

Based on the above illustration, we propose to utilize a bi-level optimization mechanism to learn $\Xi$ and $f_{gr}$, which is similar to Eq.1 and Eq.2, thus ensuring causality. Specifically, for the first level, we learn $\Xi$ and $f_{gr}$ with the support sets through the following objectives:

$$
\begin{align*}
\Xi' &\leftarrow -\alpha_1 \nabla_{\Xi} \hat{\mathcal{L}} \\
\hat{f}_{gr} &\leftarrow f_{gr} - \alpha_2 \nabla_{f_{gr}} \hat{\mathcal{L}}
\end{align*}
$$

$$
\hat{\mathcal{L}} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y^s, X^s, \Xi, f_{gr}) + \mathcal{L}_{DM}(\Xi, f_{gr})
$$

where $\mathcal{L}(Y^s, X^s, \Xi, f_{gr}) = \frac{1}{N} \sum_{i=1}^{N} y_{i,j} \log z_{i,j}^s$ and $z_{i,j}^s = h\{\text{Norm}[f_{gr}(\Xi^{T} g(x_i))] \odot [\Xi^{T} g(x_{i,j})]\}$ (7)

and for the second level, we learn $\Xi$ and $f_{gr}$ with the query sets through the following objectives:

$$
\begin{align*}
\Xi &\leftarrow -\alpha_3 \nabla_{\Xi} \hat{\mathcal{L}}' \\
\hat{f}_{gr} &\leftarrow f_{gr} - \alpha_4 \nabla_{f_{gr}} \hat{\mathcal{L}}'
\end{align*}
$$

$$
\hat{\mathcal{L}}' = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y^q, X^q, \Xi, f_{gr}) + \mathcal{L}_{DM}(\Xi, f_{gr})
$$

where $\odot$ represents the element-wise multiplication operator between two vectors, i.e., the generating representation $\Xi^{T} g(x_{i,j})$ and the weight $\text{Norm}[f_{gr}(\Xi^{T} g(x_i))]$, while $\alpha_1$, $\alpha_2$, $\alpha_3$ and $\alpha_4$ are the learning rates. Note that both in Eq.7 and Eq.8, the loss $\mathcal{L}(Y^q, X^q, \Xi, f_{gr})$ is calculated using the generating representations with causal weights instead of feature representations, which restrict the features of the samples in $\tau_i$ to be associated only with task-specific causal factors.

In summary, the learning process of $\Xi$ and $f_{gr}$ can be regarded as enforcing invariance over the support sets and the query sets, and the bi-level optimization mechanism for $\Xi$ and $f_{gr}$ can ensure causality. Meanwhile, $\Xi$ and $f_{gr}$ are learned independently with the fixed meta-learning model $f_0$ in the middle training following modularity design, thus rendering the MetaCRL a plug-and-play learner.

4.3 Overall Objective

In this subsection, we embed the above causal representation learning process into a meta-learning framework for joint optimization. The training process with MetaCRL in each batch is divided into two steps. In the first step, with $\Xi$ and $f_{gr}$ held fixed, we optimize the meta-learning model $f_{0} = h \circ g$. Specifically, the objective of the inner loop becomes:

$$
\frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y^s, X^s, f_{0})
$$

$$
\frac{1}{N} \sum_{i=1}^{N} y_{i,j} \log z_{i,j}^s
$$

where $z_{i,j}^s$ is calculated the same as Eq.7. Subsequently, the objective of the outer loop mentioned in Eq.2 becomes:

$$
\frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}(Y^q, X^q, f_{0})
$$

$$
\frac{1}{N} \sum_{i=1}^{N} y_{i,j} \log z_{i,j}^q
$$

where $z_{i,j}^q$ is calculated as mentioned in Eq.8. Next, in the second step, with the meta-learning model $f_{0}$ held fixed, we optimize $\Xi$ and $f_{gr}$ as mentioned in Eq.7 and Eq.8.

By incorporating the causal invariant-based optimization mechanism and the additional regularization term, we can effectively eliminate task confounders that lead to model degradation and improve generalization capability.

5 Experiments

In this section, we first evaluate MetaCRL on various scenarios, including sinusoid regression, image classification, drug activity prediction, and pose prediction in Subsections 5.1-5.4, respectively. Next, we conduct ablation studies and visualization in Subsections 5.5 and 5.6. Considering that MetaCRL is a plug-and-play method, we assess its performance on several meta-learning models, e.g., MAML [Finn et al., 2017], ANIL [Raghu et al., 2019], MetaSGD [Li et al., 2017], and T-NET [Lee and Choi, 2018], and multiple causal-based baselines, e.g., IFSL [Yue et al., 2020], Meta-Trans [Bengio et al., 2019], Meta-Aug [Rajendran et al., 2020], and MR-MAML [Yin et al., 2019], to demonstrate its compatibility. Considering that MetaCRL addresses the “Task Confounder” problem to enhance generalization, we also compare it with the plug-and-play generalization baselines that are most relevant to our method, i.e., MetaMix [Yao et al., 2021] and Dropout-Bins [Jiang et al., 2022]. We delay all the details of datasets, baselines, implementation details, and additional experimental results in Appendices C-F, respectively.
5.1 Sinusoid Regression

Firstly, we evaluate the performance of our MetaCRL on sinusoid regression. Following [Jiang et al., 2022], we conduct 480 tasks and the data for each task is generated in the form of $A \sin w \cdot x + b + \epsilon$, where $A \in [0.1, 5.0]$, $w \in [0.5, 2.0]$, and $b \in [0, 2\pi]$. We add Gaussian observation noise with $\mu = 0$ and $\epsilon = 0.3$ to each data point sampled from the target task. In this experiment, we set $\lambda_1$ and $\lambda_2$ to 0.4 and 0.2. We use the Mean Squared Error (MSE) as the evaluation metric.

The results are shown in Table 1. Compared to the plug-and-play baselines, MetaCRL achieves improvements with an average MSE reduction of 0.034 and 0.013, respectively. MetaCRL also demonstrates significant improvements across all the meta-learning base models, with an MSE reduction of over 0.1. Compared to the causal-based baselines, adding MetaCRL to any meta-learning model can always achieve better performance. As expected, MetaCRL exhibits significant enhancements, showcasing its high compatibility.

<table>
<thead>
<tr>
<th>Model</th>
<th>5-shot</th>
<th>10-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFSL</td>
<td>0.592 ± 0.141</td>
<td>0.178 ± 0.040</td>
</tr>
<tr>
<td>Meta-Trans</td>
<td>0.577 ± 0.123</td>
<td>0.140 ± 0.024</td>
</tr>
<tr>
<td>Meta-Aug</td>
<td>0.531 ± 0.118</td>
<td>0.103 ± 0.031</td>
</tr>
<tr>
<td>MR-MAML</td>
<td>0.581 ± 0.110</td>
<td>0.104 ± 0.029</td>
</tr>
<tr>
<td>MAML</td>
<td>0.593 ± 0.120</td>
<td>0.166 ± 0.061</td>
</tr>
<tr>
<td>MAML + MetaMix</td>
<td>0.476 ± 0.109</td>
<td>0.085 ± 0.024</td>
</tr>
<tr>
<td>MAML + Dropout-Bins</td>
<td>0.452 ± 0.081</td>
<td>0.062 ± 0.017</td>
</tr>
<tr>
<td>MAML + Ours</td>
<td>0.440 ± 0.079</td>
<td>0.054 ± 0.018</td>
</tr>
<tr>
<td>ANIL</td>
<td>0.541 ± 0.118</td>
<td>0.103 ± 0.032</td>
</tr>
<tr>
<td>ANIL + MetaMix</td>
<td>0.514 ± 0.106</td>
<td>0.083 ± 0.022</td>
</tr>
<tr>
<td>ANIL + Dropout-Bins</td>
<td>0.487 ± 0.110</td>
<td>0.088 ± 0.025</td>
</tr>
<tr>
<td>ANIL + Ours</td>
<td>0.468 ± 0.094</td>
<td>0.081 ± 0.019</td>
</tr>
<tr>
<td>MetaSGD</td>
<td>0.577 ± 0.126</td>
<td>0.152 ± 0.044</td>
</tr>
<tr>
<td>MetaSGD + MetaMix</td>
<td>0.468 ± 0.118</td>
<td>0.072 ± 0.023</td>
</tr>
<tr>
<td>MetaSGD + Dropout-Bins</td>
<td>0.435 ± 0.089</td>
<td>0.040 ± 0.011</td>
</tr>
<tr>
<td>MetaSGD + Ours</td>
<td>0.408 ± 0.071</td>
<td>0.038 ± 0.010</td>
</tr>
<tr>
<td>T-NET</td>
<td>0.564 ± 0.128</td>
<td>0.111 ± 0.042</td>
</tr>
<tr>
<td>T-NET + MetaMix</td>
<td>0.498 ± 0.113</td>
<td>0.094 ± 0.025</td>
</tr>
<tr>
<td>T-NET + Dropout-Bins</td>
<td>0.470 ± 0.091</td>
<td>0.077 ± 0.028</td>
</tr>
<tr>
<td>T-NET + Ours</td>
<td>0.462 ± 0.078</td>
<td>0.071 ± 0.019</td>
</tr>
</tbody>
</table>

Table 1: Performance (MSE) comparison on the sinusoid regression problem. “+ours” means integrating MetaCRL into the existing methods, and the best results are highlighted in **bold**.

5.2 Image Classification

Next, we conduct experiments on image classification, utilizing two benchmark datasets, i.e., miniImagenet and Omniglot. These two datasets contain 600 and 1623 tasks, respectively. We also introduce a specialized dataset called “TC”, which comprises 50 groups of tasks (300 tasks in total) identified as being affected by task confounders, i.e., tasks with negative knowledge transfer as mentioned in Subsection 3.2. More details are provided in Appendix C. In this experiment, we set $\lambda_1$ and $\lambda_2$ to 0.5 and 0.35, respectively. The evaluation metric employed here is the average accuracy.

The results are shown in Table 2. MetaCRL consistently surpasses the SOTA baselines across all datasets, indicating that it can achieve better generalization improvements than the baselines do without the need for task-specific or general-label space augmentation that the baselines need. Notably, on the “TC” dataset, MetaCRL outperforms the baselines by a significant margin, which demonstrates a unique advantage of MetaCRL in handling task confounders. In summary, MetaCRL continues to exhibit remarkable performance and adeptly eliminates task confounders.

<table>
<thead>
<tr>
<th>Model</th>
<th>Omniglot</th>
<th>miniImagenet</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFSL</td>
<td>88.51 ± 0.49</td>
<td>36.21 ± 0.62</td>
<td>\</td>
</tr>
<tr>
<td>Meta-Trans</td>
<td>87.39 ± 0.51</td>
<td>35.19 ± 1.58</td>
<td>\</td>
</tr>
<tr>
<td>Meta-Aug</td>
<td>89.77 ± 0.62</td>
<td>34.76 ± 1.52</td>
<td>\</td>
</tr>
<tr>
<td>MR-MAML</td>
<td>89.28 ± 0.59</td>
<td>35.01 ± 1.60</td>
<td>\</td>
</tr>
<tr>
<td>MAML</td>
<td>87.15 ± 0.61</td>
<td>33.16 ± 1.70</td>
<td>0.00</td>
</tr>
<tr>
<td>MAML + MetaMix</td>
<td>91.97 ± 0.51</td>
<td>38.97 ± 1.81</td>
<td>+0.42</td>
</tr>
<tr>
<td>MAML + Dropout-Bins</td>
<td>92.89 ± 0.46</td>
<td>39.66 ± 1.74</td>
<td>-0.14</td>
</tr>
<tr>
<td>MAML + Ours</td>
<td>93.00 ± 0.42</td>
<td>41.55 ± 1.76</td>
<td>+4.12</td>
</tr>
<tr>
<td>ANIL</td>
<td>89.17 ± 0.56</td>
<td>34.96 ± 1.71</td>
<td>0.00</td>
</tr>
<tr>
<td>ANIL + MetaMix</td>
<td>92.88 ± 0.51</td>
<td>37.82 ± 1.75</td>
<td>-0.10</td>
</tr>
<tr>
<td>ANIL + Dropout-Bins</td>
<td>92.82 ± 0.49</td>
<td>38.09 ± 1.76</td>
<td>+0.97</td>
</tr>
<tr>
<td>ANIL + Ours</td>
<td>92.91 ± 0.52</td>
<td>38.55 ± 1.81</td>
<td>+3.56</td>
</tr>
<tr>
<td>MetaSGD</td>
<td>87.81 ± 0.61</td>
<td>33.97 ± 0.92</td>
<td>0.00</td>
</tr>
<tr>
<td>MetaSGD + MetaMix</td>
<td>93.44 ± 0.45</td>
<td>40.28 ± 0.96</td>
<td>+0.05</td>
</tr>
<tr>
<td>MetaSGD + Dropout-Bins</td>
<td>93.93 ± 0.40</td>
<td>40.31 ± 0.96</td>
<td>+1.08</td>
</tr>
<tr>
<td>MetaSGD + Ours</td>
<td>94.12 ± 0.43</td>
<td>41.22 ± 0.93</td>
<td>+6.19</td>
</tr>
<tr>
<td>T-NET</td>
<td>87.66 ± 0.59</td>
<td>33.69 ± 1.72</td>
<td>0.00</td>
</tr>
<tr>
<td>T-NET + MetaMix</td>
<td>93.16 ± 0.48</td>
<td>39.18 ± 1.73</td>
<td>+0.28</td>
</tr>
<tr>
<td>T-NET + Dropout-Bins</td>
<td>93.54 ± 0.49</td>
<td>39.06 ± 1.72</td>
<td>+1.03</td>
</tr>
<tr>
<td>T-NET + Ours</td>
<td>93.81 ± 0.52</td>
<td>40.08 ± 1.74</td>
<td>+4.65</td>
</tr>
</tbody>
</table>

Table 2: Performance (accuracy ± 95% confidence interval) on (20-way 1-shot) Omniglot and (5-way 1-shot) miniImagenet. The “+” and “-” indicate the performance changes, and the “\textsuperscript{+}” denotes that the result is not reported. See Appendix F for full results.

5.3 Drug Activity Prediction

We also evaluate MetaCRL on drug activity prediction. pQSAR [Martin et al., 2019] is a dataset designed to forecast the activity of compounds on specific target proteins, encompassing a total of 4276 tasks. We adopt the same settings as [Yao et al., 2021] and divide the tasks into four groups. In this experiment, $\lambda_1$ and $\lambda_2$ are both set to 0.3, and the evaluation metric is the squared Pearson correlation coefficient ($R^2$), reflecting the correlation between predictions and the actual values for each task. We record both the mean and median $R^2$ values, along with the count of $R^2$ values exceeding 0.3, which stands as a reliable indicator in pharmacology.

The results are shown in Table 3. MetaCRL attains performance levels akin to the SOTA baselines across all four groups of data. Notably, we achieve a noteworthy enhancement of 3 in the reliability index $R^2 > 0.3$. The achievement of this scenario underscores the effectiveness of our MetaCRL across disparate domains and the pervasive influence of task confounders. See Appendix F for full results.

5.4 Pose Prediction

Lastly, we undertake the fourth benchmark, focusing on pose prediction. This evaluation is constructed using the Pascal 3D dataset [Xiang et al., 2014]. We randomly select 50 objects for meta-training and 15 additional objects for meta-testing. In this experiment, the values of $\lambda_1$ and $\lambda_2$ are set to 0.3 and 0.2, while the evaluation metric employed here is MSE.

The results are shown in Table 4. MetaCRL achieves the best performance. Notably, drawing insights from the findings presented in [Yao et al., 2021], we posit that augment-
Table 3: Performance comparison on drug activity prediction. “Mean”, “Med.”, and “> 0.3” are the mean, the median value of $R^2$, and the number of analyzes for $R^2 > 0.3$. The best results are highlighted in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>0.371</td>
<td>0.315</td>
<td>0.321</td>
<td>0.254</td>
</tr>
<tr>
<td>MAML + Dropout-Bins</td>
<td>0.410</td>
<td>0.376</td>
<td>0.355</td>
<td>0.257</td>
</tr>
<tr>
<td>MAML + Ours</td>
<td>0.413</td>
<td>0.378</td>
<td>0.360</td>
<td>0.261</td>
</tr>
<tr>
<td>ANIL</td>
<td>0.355</td>
<td>0.296</td>
<td>0.318</td>
<td>0.297</td>
</tr>
<tr>
<td>ANIL + MetaMix</td>
<td>0.347</td>
<td>0.292</td>
<td>0.302</td>
<td>0.258</td>
</tr>
<tr>
<td>ANIL + Dropout-Bins</td>
<td>0.394</td>
<td>0.321</td>
<td>0.338</td>
<td>0.271</td>
</tr>
<tr>
<td>ANIL + Ours</td>
<td>0.401</td>
<td>0.339</td>
<td>0.341</td>
<td>0.277</td>
</tr>
</tbody>
</table>

Table 4: Performance (MSE ± 95% confidence interval) comparison on pose prediction. More results are provided in Appendix F.

<table>
<thead>
<tr>
<th>Model</th>
<th>10-shot</th>
<th>15-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>3.113 ± 0.241</td>
<td>2.496 ± 0.182</td>
</tr>
<tr>
<td>MAML + MetaMix</td>
<td>2.429 ± 0.198</td>
<td>1.987 ± 0.151</td>
</tr>
<tr>
<td>MAML + Dropout-Bins</td>
<td>2.396 ± 0.209</td>
<td>1.961 ± 0.134</td>
</tr>
<tr>
<td>MAML + Ours</td>
<td>2.355 ± 0.200</td>
<td>1.931 ± 0.134</td>
</tr>
<tr>
<td>MetaSGD</td>
<td>2.811 ± 0.239</td>
<td>2.017 ± 0.182</td>
</tr>
<tr>
<td>MetaSGD + MetaMix</td>
<td>2.388 ± 0.204</td>
<td>1.952 ± 0.134</td>
</tr>
<tr>
<td>MetaSGD + Dropout-Bins</td>
<td>2.369 ± 0.217</td>
<td>1.927 ± 0.120</td>
</tr>
<tr>
<td>MetaSGD + Ours</td>
<td>2.362 ± 0.196</td>
<td>1.920 ± 0.191</td>
</tr>
<tr>
<td>T-NET</td>
<td>2.841 ± 0.177</td>
<td>2.712 ± 0.225</td>
</tr>
<tr>
<td>T-NET + MetaMix</td>
<td>2.562 ± 0.280</td>
<td>2.410 ± 0.192</td>
</tr>
<tr>
<td>T-NET + Dropout-Bins</td>
<td>2.487 ± 0.212</td>
<td>2.402 ± 0.178</td>
</tr>
<tr>
<td>T-NET + Ours</td>
<td>2.481 ± 0.274</td>
<td>2.400 ± 0.171</td>
</tr>
</tbody>
</table>

5.6 Visualization

To better evaluate the effect of MetaCRL, we visualize (i) knowledge transfer after using MetaCRL; and (ii) the similarity between causal factors. The former evaluates MetaCRL’s efficacy in ensuring causality and avoiding negative knowledge transfer caused by task confounders, which use the same settings as in Subsection 3.2. The latter assesses the decoupling of causal factors using cosine similarity. Figures 4 and 5 show visualizations for these two aspects, respectively. Figure 4 shows that there are almost no training tasks that lead to negative knowledge transfer with fewer iterations than Figure 1, which indicates that MetaCRL effectively eliminates task confounders. Figure 5 shows that the similarity scores between different causal factors are very low, illustrating that the disentangling module successfully decouples causal factors. More details are provided in Appendix F.

6 Conclusion

In this paper, we discover a valuable problem called “Task Confounder”, and propose a novel method called MetaCRL to address its unique challenges. We begin by analyzing a counterintuitive negative knowledge transfer phenomenon with SCM, revealing spurious correlations between causal factors of the training tasks and the label space, i.e., “Task Confounder”. Then, we propose MetaCRL, which consists of two modules: (i) a disentangling module that acquires generating factors and eliminates task confounders; and (ii) a causal module that ensures causality of the obtained generating factors. It is a plug-and-play causal representation learner that can be applied to any meta-learning baseline. Extensive experiments demonstrate the effectiveness and robustness of MetaCRL. Our work uncovers a novel and significant issue in ML, providing valuable insights for future research.
Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the Postdoctoral Fellowship Program of CPSF No. 2023M743639, and the Special Research Assistant Fund, Chinese Academy of Sciences No. E3YD590101. The Appendix is provided in https://arxiv.org/abs/2312.05771.

Contribution Statement

Jingyao Wang and Yi Ren made equal contributions. All the authors participated in designing research, performing research, analyzing data, and writing the paper.

References


[Qiang et al., 2023] Wenwen Qiang, Jiangmeng Li, Bing Su, Jianlong Fu, Hui Xiong, and Ji-Rong Wen. Meta


