MINIMIZING MEMORIZATION IN META-LEARNING: A CAUSAL PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

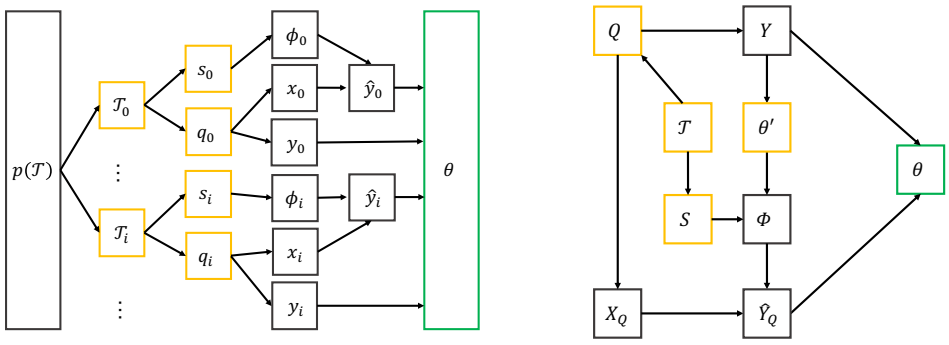
Meta-learning has emerged as a potent paradigm for quick learning of few-shot tasks, by leveraging the meta-knowledge learned from meta-training tasks. Well-generalized meta-knowledge that facilitates fast adaptation in each task is preferred; however, recent evidence suggests the undesirable memorization effect where the meta-knowledge simply memorizing all meta-training tasks discourages task-specific adaptation and poorly generalizes. There have been several solutions to mitigating the effect, including both regularizer-based and augmentation-based methods, while a systematic understanding of these methods in a single framework is still lacking. In this paper, we offer a novel causal perspective of meta-learning. Through the lens of causality, we conclude the universal label space as a confounder to be the causing factor of memorization and frame the two lines of prevailing methods as different deconfounder approaches. Remarkably, derived from the causal inference principle of front-door adjustment, we propose two frustratingly easy but effective deconfounder algorithms, i.e., sampling multiple versions of the meta-knowledge via Dropout and grouping the meta-knowledge into multiple bins. The proposed causal perspective not only brings in the two deconfounder algorithms that surpass previous works in four benchmark datasets towards combating memorization, but also opens a promising direction for meta-learning.

1 INTRODUCTION

Recently, there has been renewed interest in meta-learning which empowers more human-like machines that suffice to learn a wide range of tasks with minimal supervision (Bengio et al., 1991; Thrun & Pratt, 2012; Finn et al., 2017; Raghu et al., 2020). While metric-based meta-learning algorithms (Vinyals et al., 2016; Snell et al., 2017) only solve few-shot classification problems, we focus on gradient-based meta-learning algorithms in this work that are more flexible (Finn et al., 2017; Li et al., 2017). Gradient-based meta-learning algorithms formulate the meta-knowledge as the initialization for a base learner and learn the initialization by a bi-level optimization procedure during the meta-training phase. Concretely, the initialization is adapted to each meta-training task by its support set, while the performance of the adapted model on its query set in turn serves as feedback to update the initialization.

This bi-level optimization scheme, though designed to learn a well-generalized initialization, runs a high risk of inducing a sufficiently expressive initialization that memorizes all meta-training tasks. This kind of overfitting is named *memorization overfitting* (Yin et al., 2020), where the initialization solves the query set even without relying on the support set for adaptation. As a consequence, such an initialization poorly generalizes to meta-testing tasks. As suggested in Yin et al. (2020), the more non-mutually exclusive meta-training tasks are and the more powerful the model initialization is, the higher risk of memorization arises. To combat the memorization overfitting, Yin et al. (2020) proposed to regularize the capacity of the initialization, and task augmentation strategies have been recently explored in (Rajendran et al., 2020; Yao et al., 2021).

Despite the effectiveness of the three algorithms, understanding their benefits rigorously within a unified analytic tool remains a mystery. To bridge the gap, we develop a causal perspective on meta-learning, as illustrated by the causal graph in Figure 1. We argue that the universal label space of the base learner turns to be a confounder causing a *spurious correlation* between the initializations learned in different steps of meta-training. Such a spurious correlation biases the meta-knowledge



(a) Workflow of meta-learning

(b) Causal graph of meta-learning

Figure 1: An overview of meta-learning training. Yellow nodes are inputs to the meta-model, green nodes are outputs by the meta-model, and black nodes represent intermediate variables. (a) The workflow of meta-learning, where a set of meta-training tasks are sampled from $p(\mathcal{T})$, and a task-specific model ϕ_i is updated by the support set s_i, then meta-knowledge θ is optimized on the likelihood of query sets $\{q_i\}_{i=1}^N$. (b) The causal graph of meta-learning, where θ' and θ are the initialization learned from last step and current step, respectively.

that should be only updated by the performance of task-specific models. Fortunately, the causal graph in Figure 1b offers valuable insights into how to minimize memorization via deconfounder approaches. In particular, we have demonstrated the deconfounding role of both lines of existing works: 1) regularizer-based methods directly weaken the correlation between the initialization meta-trained in the last step (*i.e.*,θ') and the task-specific model Φ during meta-training, though the limited flexibility of the initialization in this case still promotes spurious relations; 2) augmentation-based methods take different mapping functions from task labels to the universal label space for various tasks, but the performance highly depends on the independence between mapping functions.

Drawing upon the causal perspective, we put forward a new direction of deconfounder approaches by applying the causal inference principle of front-door adjustment. We propose two easy implementations of this principle, which are to sample multiple stratification of the initialization by Dropout and to predict the label as well as the bin that the label belongs to, respectively. Take the backbone of MAML (Finn et al., 2017) as an example. We name the two deconfounder approaches as MAML-Dropout and MAML-Bins, respectively.

The main contributions of our paper are as follows. (1) We, for the first time, develop a causal perspective of meta-learning and shed light on the memorization overfitting with causality in Section 2.2. (2) We place existing methods into the proposed causal framework and adequately demonstrate how they alleviate the memorization overfitting in Section 2.3 and Section 2.4. (3) We propose a new deconfounder approach following the principle of front-door adjustment in Section 2.4 and two methods that implement the approach in Section 3. (4) We showcase that our methods remain compatible with off-the-shelf meta-learning algorithms and consistently improve their performance.

2 Problem Formulation

2.1 Meta-Learning and the Overfitting Problem

Meta-learning learns the model initialization θ from a series of tasks $\mathcal{T}_i$ sampled from a specific task distribution $p(\mathcal{T})$. All tasks in $p(\mathcal{T})$ share some common features, so that starting from the initialization θ a new task sampled from the same task distribution can be quickly learned with a resulting task-specific model ϕ. The tasks used to learn the initialization are considered as meta-training tasks D_{train}, while novel tasks are meta-testing tasks D_{test}. Each i-th task $\mathcal{T}_i$ consists of a support set $s_i = \{(x_{i,j}^s, y_{i,j}^s)\}_{j=1}^{K_i^s}$ and a query set $q_i = \{(x_{i,j}^q, y_{i,j}^q)\}_{j=1}^{K_i^q}$, where (x, y) denote the features and the label of a sample, K_i^s and K_i^q denote the number of support and query samples.

Gradient-based meta-learning formulate learning such a initialization θ as a bi-level optimization problem (See Figure 1a). During *inner-loop optimization*, the adapted model ϕ_i for the i-th task is initialized from θ and updated by its support set s_i. In outer-loop optimization, the initial-

ization θ is optimized according to performances of adapted models on query sets, *i.e.*,by losses between label $y^q_{i,j}$ and prediction $\hat{y}^q_{i,j}$. Following (Grant et al., 2018; Gordon et al., 2019; Yin et al., 2020), we formulate the objective of meta-learning as maximizing the conditional likelihood $p_\phi(\hat{y}^q|x^q, \theta, s)$, where the *inner-loop* optimization learns the conditional distribution of task-specific models $p(\phi|\theta, s)$ and the *outer-loop* optimizes the distribution of θ $p(\theta|D_{train})$. Consequently, the objective for *inner-loop* optimization (*i.e.*, task objective) is

$$\mathcal{L}(\phi_i) = \frac{1}{K^s}\sum_{j=1}^{K^s}\mathcal{L}(f_{\phi_i,\theta}(x^s_{i,j}), y^s_{i,j}),$$

and the objective for *outer-loop* optimization (*i.e.*, meta-objective) is

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{p(\phi_i|\theta,s_i)}[\frac{1}{K^q_i}\sum_{j=1}^{K^q_i}\mathcal{L}(f_{\phi_i,\theta}(x^q_{i,j}), y^q_{i,j})].$$

Take the algorithm of MAML (Finn et al., 2017) as a concrete example. During the outer-loop update, MAML optimizes the delta function $p(\theta|D_{train})$ on meta-training tasks to learn the initialization θ of a base learner f; in the inner-loop update, $p(\phi|\theta, s)$ is also a point estimation by gradient optimization, *i.e.*,$\phi_i = \theta - \alpha\nabla_\theta\frac{1}{K^s}\sum_{j=1}^{K^s}\mathcal{L}(f_\theta(x^s_{i,j}), y^s_{i,j})$. Finally, we predict for a query sample by the adapted model, *i.e.*, $p(\hat{y}^q_{i,j}|x^q_{i,j}, \phi_i) = f_{\phi_i}(x^q_{i,j})$ (Grant et al., 2018; Yin et al., 2020).

In meta-learning, there are two types of overfitting problems: 1) memorization overfitting, which happens when the meta-knowledge memorizes all query sets in meta-training tasks even without adapting on the support sets, and 2) learner overfitting, which happens when meta-knowledge is only effective on meta-training tasks and fails to generalize to meta-testing tasks (Yin et al., 2020; Rajendran et al., 2020). In this paper, we focus on the former. Regularizer-based and augmentation-based methods have been proposed to combat the memorization overfitting, but how to systematically understand the benefits of these methods within a unified analytic tool is still a mystery.

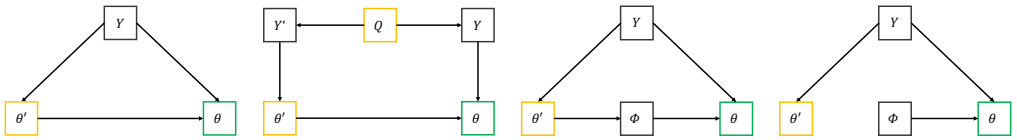
2.2 A Causal View of Meta-Learning

Firstly, we introduce causality and causal graph, which are main theories supporting our work. Then, we show the causal graph for gradient-based meta-learning and formulate the memorization overfitting (Yin et al., 2020; Rajendran et al., 2020) in a causal view. Besides, we explain the reason why existing methods alleviate the memorization in various degrees via our causal graph. Lastly, we propose a deconfounding principle with frontdoor adjustment.

Causation and Causal Graph. Causation describes causal relationship among variables instead of correlation. Causal graph (Pearl et al., 2016) addresses causality problems with a directed acyclic graph $G =< V, E >$, where a node $V_i \in V$ denotes a variable and a directed edge $V_i \to V_j \in E$ denotes that variable V_i is a direct cause of V_j.

Revisit of meta-learning: a causal view. During the meta-training phase, given N meta-training tasks, we define s_i and q_i to be the support set and the query set of task i, respectively. Then, in meta-training data D_{train}, we have all support sets S as $\{s_i\}_{i=1}^N$ and all query sets Q as $\{q_i\}_{i=1}^N$. Support sets S and query sets Q consist of randomly drawn examples that are *i.i.d.* Thus, given meta-training tasks, S and Q are independent. Given query sets of the training set, the input variables $X_Q = \{x_{q_i}\}_{i=1}^N$ is determined, so $\boldsymbol{Q} \to \boldsymbol{X_Q}$. It is obvious that query sets Q has a causal effect on labels of query sets (*i.e.*,$\boldsymbol{Q} \to \boldsymbol{Y}$). According to the workflow of meta-learning shown in Figure 1a, we can easily find the causal links of inner-loop optimization $\boldsymbol{S} \to \boldsymbol{\Phi}$ and outer-loop optimization $\boldsymbol{\Phi} \to \boldsymbol{\hat{Y}_Q} \to \boldsymbol{\theta} \leftarrow \boldsymbol{Y}$ as shown in Figure 1b.

In Figure 1b, let θ' denote the meta-knowledge learned from last step. It is trained in the same way as θ (the meta-knowledge learned from current step); therefore, there is also a causal link from Y to θ' (i.e., $\boldsymbol{Y} \to \boldsymbol{\theta'}$). We omit the connection from the predictions in the last step since the causal effect can be merged into the effect from $\boldsymbol{\Phi} \to \boldsymbol{\hat{Y}_Q}$ (See Appendix A.1). The connection $\boldsymbol{\theta'} \to \boldsymbol{\Phi}$ denotes that meta-knowledge θ' obtained in last update has a causal effect on task-specific models Φ since Φ is always trained by leveraging meta-knowledge θ' as initialization. Finally, we can obtain the causal graph of meta-learning as shown in Figure 1b.



(a) (b) (c) (d)

Figure 2: Simplified causal graphs of meta-training and deconfounded methods. (a) simplified causal graph among Y, θ' and θ. (b) deconfounded meta-knowledge. (c) simplified causal graph among Y, θ', Φ and θ. (d) deconfounded by frontdoor adjustment. See Figure 4 in Appendix A.2 for complete causal graphs.

In Figure 1b, the key idea of meta-learning is that by utilizing the past meta-knowledge θ' as initialization, one can optimize the task-specific model Φ with new support sets S for a more general meta-knowledge θ. But it ignores the confounder Y (in causal path $\hat{Y}_Q \leftarrow \Phi \leftarrow \theta' \leftarrow Y \rightarrow \theta$) which influences both the meta-knowledge in the past step and the current step, leading to *spurious correlation* between θ' and θ. This *spurious correlation* biased by the confounder Y makes the task-specific model especially challenging to be sufficiently adapted by the support set S, thereby putting the initialization at high risk of memorization.

We would highlight the difference of labels Y in meta-learning from those in conventional machine learning. In meta-learning, despite the universal label space, the same label varies from task to task in semantic meanings. For example, the label of 0 may indicate "dog" in one task and represent "cat" in another. Thus, Y is not only affected by query sets Q, but also by a hidden variable (i.e., how to map a task label to the universal label space for various tasks). The hidden variable can be denoted as a unobserved exogenous variable and omitted in Figure 1b.

Deconfounded meta-learning. In meta-learning, the meta-knowledge θ is learned by $p(\theta|\theta', S, Q)$ in each step, although the correlation between θ' and θ would be spurious since the causal path $\hat{Y}_Q \leftarrow \Phi \leftarrow \theta' \leftarrow Y \rightarrow \theta$ shown in Figure 1b demonstrates Y is a confounder of path $\theta' \rightarrow \cdots \rightarrow \theta$ (see proof in A.2). Given $D_{train} = (S, Q)$ and omitting the intermediate nodes, we simplify the causal graph with three nodes $\{Y, \theta', \theta\}$ as shown in Figure 2a. Y opens the backdoor path from θ and θ'. However, the backdoor adjustment is not applicable to to cut the causal relationship between Y and θ' because the edges $\boldsymbol{Y \rightarrow \theta}$ and $\boldsymbol{Y \rightarrow \theta'}$ in the causal graph would be changed simultaneously since they perform exactly the same roles in meta-learning. Despite this, we propose two kinds of deconfounded methods applying to MAML (Finn et al., 2017) —one is inspired by some recent works (Rajendran et al., 2020; Yin et al., 2020; Yao et al., 2021; Tseng et al., 2020); the other is based on front-door criterion. These methods are introduced in Section 2.3 and Section 2.4.

2.3 Deconfounded Meta-knowledge

One possible solution to break the connection from Y to θ' is to use different label mapping functions in different steps of meta-training as shown in Figure 2b. $\boldsymbol{Y' \leftarrow Q \rightarrow Y}$ denotes two kinds of meta label representation of query sets' labels. In this fork structure, Y and Y' are independent, conditional on query sets Q. The backdoor path from Y to θ' is closed. Thus, there is no confounder in the new causal graph and the model can learn $p(\theta|\theta', S, Q)$ directly.

Under this view, Meta-augmentation (Rajendran et al., 2020) and MetaMix (Yao et al., 2021) can be considered as two deconfounded meta-knowledge models. Both these two methods randomize the labels of query sets to prevent the memorization. As spurious correlations are reduced, these methods achieve better performance than original MAML. Meta-augmentation applies a CE-Increasing augmentation(Rajendran et al., 2020) in each step which changes the labels of the same task. MetaMix generates fake data with manifold mixup (Verma et al., 2019) and channel shuffle. As a result, in different steps, meta-knowledge θ is even optimized on a different label space.

In fact, even these two methods reduce spurious correlations, they have different performances on same tasks. This phenomenon is due to only a partial of the correlation is blocked by conditioning on Q. The augmentation function sampled from a random space still confounds the model. To be specific, Y' and Y are not independent.

Another possible way to deconfound meta-knowledge is to constrain the meta-knowledge θ' to weaken the correlation between Y and θ'. Meta-regularization on weights (Yin et al., 2020) ap-

plies this method. Meta-regularization limits the meta-knowledge by a meta-regularized objective to avoid memorization of meta-training tasks. However, naive regularization weakens the usability of fast adaptation in the inner-loop as $Y \rightarrow \theta$ is limited simultaneously. As a result, regularizer-based method suffers from a *trade-off* of effectiveness and generalization. Besides, the weakened correlation between θ' and Φ still confounds the model.

2.4 DECONFOUNDED META-LEARNING MODEL

With considering the mediator Φ in the path from θ' to θ, we can also simplify the causal path $\hat{Y}_Q \leftarrow \Phi \leftarrow \theta' \leftarrow Y \rightarrow \theta$ in Figure 1b with four nodes $\{Y, \theta', \Phi, \theta\}$ as shown in Figure 2c. Then, the other way block the backdoor path from θ' to θ is to disconnect the path from meta-knowledge θ' to Φ via frontdoor adjustment. We propose a novel way to deconfound meta-learning model in Figure 2d, and propose to calculate $p(\theta|do(\Phi), Q)$ instead of $p(\theta|\Phi, Q)$, which enables the model eliminating the confounder Y, *i.e.*,$p(\theta|do(\theta'), S, Q)$. This is so called "frontdoor adjustment", which is proved in Appendix A.2. The deconfounded meta-learning model is

$$\begin{aligned} p(\theta|do(\theta'), S, Q) &= \sum_{\Phi} p(\Phi|\theta', S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) = \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i), \end{aligned} \tag{1}$$

where $p(\Phi|\theta', S) = 1$ in delta function. In Eq.(1), we stratify the confounded past meta-knowledge θ', *i.e.*,$\theta' = \{\theta'_i\}$, where θ'_i is a stratum of θ'. $p(\theta|\Phi, \theta'_i, Q)$ denotes optimizing θ grouped by θ'_i. Thus, Φ is grouped in the same way. We propose two implementations of MAML to stratify θ' in Section 3. After frontdoor adjustment, we break the frontdoor path from θ' to Φ. Therefore, the model would not memorize the query-set of meta-training tasks.

3 TWO METHODS TO DECONFOUND MAML

3.1 MAML-DROPOUT

Our first idea is inspired from MC-dropout (Gal & Ghahramani, 2016). We split θ' into different parts by dropout, *i.e.*,

$$p(\theta|do(\Phi), Q) = \int p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) d\theta'_i \approx \frac{1}{M} \sum_{i=1}^{M} p(\theta|\Phi, \theta'_i, Q) = \frac{1}{M} \sum_{i=1}^{M} p(\theta|\Phi, \theta', Q, z_i), \tag{2}$$

where M is sample times, θ'_i indicates a combination of θ' and z_i, which is a set of dropout variables sampled from Bernoulli distribution. θ' is independent with z_i.

Different from DropGrad (Tseng et al., 2020), we add dropout layers in forward network only on query sets during meta-training. We adopt multi-step optimization in inner-loop (Antoniou et al., 2019) to update almost all meta-knowledge on a training step, which avoids limiting the model's flexibility. In each training step, a batch of meta-training tasks are used to optimize the model, so empirically, we sample different parts of θ' through Monte Carlo method for different tasks in a batch to approximate results of Eq.(2). The meta-training objective of MAML-Dropout is

$$\begin{aligned} &\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} \frac{v_t}{K_i^q} \sum_{j=1}^{K_i^q} \mathcal{L}(f_{\phi_{i,t}, z_{i,t}}(x_j^q), y_j^q), \\ &s.t.\ \phi_{i,t} = \begin{cases} \theta', & \text{if } t = 0 \\ \phi_{i,t-1} - \alpha \nabla_{\phi_{i,t-1}} \frac{1}{K_i^s} \sum_{j=1}^{K_i^s} \mathcal{L}(f_{\phi_{i,t-1}}(x^s), y^s)), & \text{else} \end{cases}, \end{aligned} \tag{3}$$

where N is the number of meta-training tasks, T is the number of inner-loop steps, v_t denotes the importance weight of the target set loss at step t (Antoniou et al., 2019), K_i^q denotes the number of query samples in the i-th task, learned weights $\phi_{i,t}$ and random variable $z_{i,t}$ parameterizes the adaptive model. The gradients of the dropped part in $\phi_{i,t}$ are set to zero.

In the meta-testing phase, we remove all dropout layers. All meta-knowledge guides the model to learn a new task without regularization and *spurious correlation*.

3.2 MAML-BINS

As meta-knowledge in MAML extracts powerful features (Raghu et al., 2020), we propose another method to stratify θ' through the linear combination of features. In this situation, we add an auxiliary task to classify training data to several bins covering all training data points and dividing features into finite groups. So we have Eq.(1) as:

$$p(\theta|do(\Phi), Q) = \frac{1}{M}\sum_{M} p(\theta|\Phi, \theta'_m, Q) = \frac{1}{M}\sum_{M} p(\theta|\Phi, \theta', Q, W_m), \tag{4}$$

where $\theta'_m = W_m \times \theta'$ is a linear combination of features determined by a bin m.

We define an N-way M-bin task as an N-way classification with M bins. We propose an auxiliary task as a M-class classification problem to assign data to different bins. We detail the auxiliary task in Appendix C. We cluster features of all training data to K bins with a pretrained classifier and set the cluster id of the data point as its auxiliary task label, *i.e.*,b. During the inner-loop step, the model learns the main task and the auxiliary classification, *i.e.*,the combination of a group of meta-knowledge. The output of model is $O = f_\phi(x)$, where $O \in \mathbb{R}^{M\times N}$. The prediction of bins is $p(\hat{b}|x, \phi) = \frac{1}{N}\sum_{j=1}^{N} O_j^T$ and the prediction of classification is $p(\hat{y}|x, \phi) = \frac{1}{M}\sum_{i=1}^{M} O_i$, where O_j^T is the j-th column vector of O and O_i is the i-th row vector of O, which is parameterized by $W_i \times \phi$. Therefore, we have the outer-loop objective is:

$$\mathcal{L}(\theta) = \frac{1}{N_T}\sum_{i=1}^{N_T} \mathbb{E}_{p(\phi_i|\theta,s_i)}\{\frac{1}{K_i^q}\sum_{j=1}^{K_i^q}[\mathcal{L}(\hat{y}_j^q, y_j^q) + \lambda\mathcal{L}(\hat{b}_j^q, b_j^q)]\}, \tag{5}$$

where λ is the weight of bin-classification loss.

As for a regression task, we split the value range of training data into several intervals as bins and set the interval id of the data point as its auxiliary task label. The training process is like a "1-way M-bin classification", and its objective is same as Eq.(5). A M-bin task reshapes meta-knowledge θ' to M stratifications, and frontdoor adjustment is implemented in this way.

MAML-Dropout only adds dropout layers and MAML-Bins only adds an additional objective in the outer-loop optimization, which are easy to apply and do not incur additional computation overhead. We combine MAML-Dropout and MAML-Bins in Algorithm 1 and discuss more in Appendix B.

4 RELATED WORK

Gradient-based meta-learning methods (Finn et al., 2017; Raghu et al., 2020; Grant et al., 2018; Li et al., 2017; Lee & Choi, 2018) learn a model initialization as meta-knowledge and fast adapt to new tasks with the initialization. Because these methods are model agnostic, they are wildly implemented in many research areas, *e.g.*,few-shot learning, reinforcement learning and transfer-learning. However, the learned initialization tends to overfit the meta-training tasks, especially the query set of meta-training tasks. Yin et al. (2020) and Rajendran et al. (2020) firstly formulated the meta-overfitting problem. Various methods were proposed to solve the overfitting problem in gradient-based methods. The most common way is using the standard regularization techniques, such as adding adaptive noise (Lee et al., 2019), limiting trainable parameters (Yin et al., 2020; Oh et al., 2021; Zintgraf et al., 2019), and enforcing the similarity of different tasks (Jamal & Qi, 2019).A regularizer-based method, DropGrad (Tseng et al., 2020), applies dropout to support sets but not to the query set. The common positive effect of these methods is weakening the *spurious correlation*. But, they still use confounded past meta-knowledge which limits the flexibility of meta-knowledge as described in Section 2.3. Recently, Yao et al. (2021) and Rajendran et al. (2020) proposed task-augmentation methods to solve the overfitting problem. Both these methods partially block the backdoor path and achieve outstanding performance than regularization algorithms.

Our work solves the confounder of meta-learning through causal reasoning, especially causal graph and *do*-calculus (Pearl, 2009; Pearl et al., 2016). Some recent works (de Haan et al., 2019; Zhang et al., 2020; Kocaoglu et al., 2018; Yang et al., 2020; Kurutach et al., 2018; Qi et al., 2020; Nair et al., 2019; Mahajan et al., 2019; Nauta et al., 2019) have shown that causal reasoning helps deep learning models to mine causal relation instead of correlation; meanwhile, the deep model's powerful representation ability is beneficial to causal models dealing with high-dimensional data. Most relevant

to ours are Bengio et al. (2020) and Yue et al. (2020) which combined meta-learning and causality. The goal of Bengio et al. (2020) is to leverage a meta-learning objective to discover causal structures for fast transfer learning, which solves a completely different research problem from ours. Yue et al. (2020) proposed IFSL to deconfound the pre-trained knowledge during meta-testing, which cannot handle the memorization issue arised during meta-training. Our work mainly focuses on the meta-training process and solves memorization overfitting, which is crucial in meta-learning.

5 EXPERIMENT

We compare our methods with the state-of-the-art solution of memorization overfitting —MetaMix (Yao et al., 2021). We evaluate the performance on different backbones, such as MAML (Finn et al., 2017), ANIL (Raghu et al., 2020), MetaSGD (Li et al., 2017), and T-NET (Lee & Choi, 2018) (together with MetaMix in Appendix E.4), to show the compatibility of our methods. In addition, the ablation study and the analysis of hyperparameters show the robustness of our methods.

5.1 SINUSOID REGRESSION

First, we evaluate the performance on a toy sinusoid regression problem. We construct a more challenging problem to further corroborate the superiority of our methods. The data for each task is created in forms of $A \cdot \sin w \cdot x + b + \epsilon$, with $A \in [0.1, 5.0]$, $w \in [0.5, 2.0]$ and $b \in [0, 2\pi]$. Gaussian observation noise with $\mu = 0$ and $\epsilon = 0.3$ is added to each data point sampled from the target task. The regression results are computed by a two-layer Multilayer Perceptron. Implementation of our methods (MAML-Dropout+MAML-Bins) in this experiment uses 5 bins and 0.3 dropout rate. Please kindly refer to Appendix D.1 for more details of the experimental setup. We report the mean squared error (MSE) as the evaluation criterion.

Table 1: Performance (MSE $\pm$ 95% confidence interval) of sinusoid regression problem.

Model	5-shot	10-shot
IFSL	0.59 ± 0.15	0.15 ± 0.04
DropGrad	0.57 ± 0.15	0.14 ± 0.07
MR-MAML	0.57 ± 0.11	0.10 ± 0.02
Meta-Aug	0.53 ± 0.10	0.10 ± 0.02
ANIL	0.54 ± 0.10	0.10 ± 0.02
ANIL-MetaMix	0.51 ± 0.10	0.08 ± 0.02
ANIL-ours	$\mathbf{0.49 \pm 0.10}$	$\mathbf{0.08 \pm 0.02}$
MAML	0.59 ± 0.12	0.16 ± 0.06
MAML-MetaMix	0.47 ± 0.10	0.08 ± 0.02
MAML-ours	$\mathbf{0.45 \pm 0.08}$	$\mathbf{0.06 \pm 0.01}$
MetaSGD	0.56 ± 0.11	0.14 ± 0.04
MetaSGD-MetaMix	0.46 ± 0.10	0.07 ± 0.02
MetaSGD-ours	$\mathbf{0.43 \pm 0.07}$	$\mathbf{0.04 \pm 0.01}$
T-Net	0.54 ± 0.11	0.11 ± 0.03
T-Net-MetaMix	0.49 ± 0.10	0.08 ± 0.02
T-Net-ours	$\mathbf{0.47 \pm 0.09}$	$\mathbf{0.07 \pm 0.02}$

According to Table 1, comparing with some basic gradient-based meta-learning algorithms: IFSL (Yue et al., 2020) only focuses on the meta testing phase, so, it cannot improve the performance in a non-mutually-exclusive task; MR-MAML (Yin et al., 2020) achieves a minor improvement, which accords with our analysis in Section 2.3. Meta-Augmentation's (Rajendran et al., 2020) error is larger than MetaMix (Yao et al., 2021) because mapped labels in different steps generated by MetaMix is more random and independent. Our methods bring a huge improvement applied in different baselines.

We also evaluate our method separately, as shown in Figure 3b. We find that both methods have positive effects and using them together achieves the best performance. Besides, we explore how the number of bins and the dropout rate influence results (see Figure 5b and Figure 5d in Appendix E.1). As a result, the dropout rate should be in $[0.1, 0.3]$ because a low dropout rate stratifies θ' in similar ways while a high dropout rate limits the generalization of meta-knowledge. For the same reason, the optimal number of bins M is also in a range of $[4, 10]$.

5.2 DRUG ACTIVITY PREDICTION

Following Yao et al. (2021), we apply our methods to the drug activity prediction task (Martin et al., 2019). The task set contains 4276 assays (*i.e.*,tasks). In each task, we need to predict activities of several compounds on a specific target protein; whereas there are only a few labeled data in the support set. We split tasks into meta-training tasks, meta-validation tasks and meta-testing tasks in the same way as Yao et al. (2021). Other details of datasets and settings are given in Appendix D.3.

Table 2: Performance of drug activity prediction.

Model	Group 1			Group 2			Group 3			Group 4		
	Mean	Med.	>0.3	Mean	Med.	>0.3	Mean	Med.	>0.3	Mean	Med.	>0.3
ANIL	0.357	0.294	50	0.300	0.245	45	0.327	0.301	50	0.338	0.302	50
ANIL-ours	0.394	0.321	53	0.312	0.284	46	0.338	0.271	48	0.370	0.297	50
MAML	0.366	0.317	53	0.312	0.239	44	0.321	0.258	43	0.348	0.280	47
MAML-ours	0.410	0.376	60	0.320	0.275	46	0.355	0.257	48	0.370	0.337	56
MetaSGD	0.388	0.306	51	0.298	0.236	41	0.326	0.237	46	0.353	0.316	52
MetaSGD-ours	0.390	0.342	57	0.316	0.269	43	0.358	0.339	56	0.360	0.311	50

We also evaluate the square of Pearson coefficient R^2 between the prediction and the ground-truth in each task (Martin et al., 2019; Yao et al., 2021), rather than the mean squared error as an evaluation metric because values of these data are noisy and the coefficient is more meaningful. As the same reason, MAML-Bins brings additional noise, so we only apply MAML-Dropout with 0.1 dropout rate in this experiment. Results of our method are reported in Table 2. In four different groups, our method (MAML-Dropout) is capable of improving the performance on all three backbones.

5.3 Pose Prediction

We also evaluate another regression task created from Pascal 3D data (Xiang et al., 2014). Following Yin et al. (2020), we randomly select 50 objects for meta-training and the other 15 objects for meta-testing. Same as the past works (Yin et al., 2020), we use a base model with a three-convolution-block encoder and a four-convolution-block decoder. Implementation of our methods (MAML-Dropout+MAML-Bins) in this experiment uses 5 bins and 0.2 dropout rate. Detailed settings are described in Appendix D.2.

Table 3: Performance (MSE $\pm$ 95% confidence interval) of pose prediction.

Model	10-shot	15-shot
Weight Decay	2.772 ± 0.259	2.307 ± 0.226
CAVIA	3.021 ± 0.248	2.397 ± 0.191
Meta-dropout	3.236 ± 0.257	2.425 ± 0.209
Meta-Aug	2.553 ± 0.265	2.152 ± 0.227
MR-MAML	2.907 ± 0.255	2.276 ± 0.169
IFSL	3.186 ± 0.256	2.482 ± 0.231
TAML	2.785 ± 0.261	2.196 ± 0.163
ANIL	6.746 ± 0.416	6.513 ± 0.384
ANIL-MetaMix	6.354 ± 0.393	6.112 ± 0.381
ANIL-ours	$\mathbf{6.289 \pm 0.416}$	$\mathbf{6.064 \pm 0.397}$
MAML	3.098 ± 0.242	2.413 ± 0.177
MAML-MetaMix	2.438 ± 0.196	2.003 ± 0.147
MAML-ours	$\mathbf{2.396 \pm 0.209}$	$\mathbf{1.931 \pm 0.134}$
MetaSGD	2.803 ± 0.239	2.331 ± 0.182
MetaSGD-MetaMix	2.390 ± 0.191	1.952 ± 0.154
MetaSGD-ours	$\mathbf{2.369 \pm 0.204}$	$\mathbf{1.926 \pm 0.112}$
T-Net	2.835 ± 0.189	2.609 ± 0.213
T-Net-MetaMix	2.563 ± 0.201	2.418 ± 0.182
T-Net-ours	$\mathbf{2.487 \pm 0.212}$	$\mathbf{2.402 \pm 0.178}$

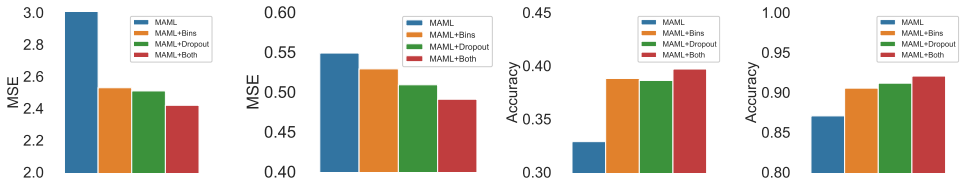
In Table 3, we evaluate more algorithms in this experiment. We find it is difficult for regularizer-based methods to overcome memorization overfitting, especially under 10-shot setting. If there are only few samples in the support set, model is hard to adapt to a specific task and tends to memorize the query set in the meta trainging phase. In this task, our methods' performance exceeds MetaMix again, which highlights the deconfounding ability of our methods. In Figure 3a, applying our methods separately, the performance of model still achieves a significant advancement.

5.4 Image Classification

We also study the memorization overfitting in a few-shot image classification problem with two benchmarks, Omniglot (Lake et al., 2011) and MiniImagenet (Vinyals et al., 2016). Following Yin et al. (2020); Rajendran et al. (2020), these experiments are under a non-mutually-exclusive setting. "non-mutually-exclusive N-way K-shot classification" means each class is assigned with an unchangeable label from 1 to N in different tasks and training steps. Each task contains N classes labeled from 1 to N. This setting aggravates the memorization overfitting according to the causality described in Section 2.3 and highlights the power of deconfounding. We use a four-block convolutional network, which is the same as the model of Yao et al. (2021) and suffer from less meta-overfitting than the deeper network used in (Yin et al., 2020; Rajendran et al., 2020). We evaluate

Table 4: Performance (accuracy $\pm$ 95% confidence interval) of image classification on Omniglot and MiniImagenet.

Model	Omniglot		MiniImagenet	
	20-way 1-shot	20-way 5-shot	5-way 1-shot	5-way 5-shot
Weight Decay	$86.81 \pm 0.64\%$	$96.20 \pm 0.17\%$	$33.19 \pm 1.76\%$	$52.27 \pm 0.96\%$
CAVIA	$87.63 \pm 0.58\%$	$94.16 \pm 0.20\%$	$34.27 \pm 1.79\%$	$50.23 \pm 0.98\%$
DropGrad	$87.69 \pm 0.57\%$	$94.21 \pm 0.20\%$	$34.42 \pm 1.70\%$	$52.92 \pm 0.98\%$
MR-MAML	$89.28 \pm 0.59\%$	$96.66 \pm 0.18\%$	$35.00 \pm 1.60\%$	$54.39 \pm 0.97\%$
Meta-dropout	$85.60 \pm 0.63\%$	$95.56 \pm 0.17\%$	$34.32 \pm 1.78\%$	$52.40 \pm 0.96\%$
TAML	$87.50 \pm 0.63\%$	$95.78 \pm 0.19\%$	$33.16 \pm 1.68\%$	$52.78 \pm 0.97\%$
ANIL	$88.35 \pm 0.56\%$	$95.85 \pm 0.19\%$	$34.13 \pm 1.67\%$	$52.59 \pm 0.96\%$
ANIL-MetaMix	$92.24 \pm 0.48\%$	$98.36 \pm 0.13\%$	$37.94 \pm 1.75\%$	$59.03 \pm 0.93\%$
ANIL-ours	$\mathbf{92.82 \pm 0.49\%}$	$\mathbf{98.42 \pm 0.14\%}$	$\mathbf{38.09 \pm 1.76\%}$	$\mathbf{59.17 \pm 0.94\%}$
MAML	$87.40 \pm 0.59\%$	$93.51 \pm 0.25\%$	$32.93 \pm 1.70\%$	$51.95 \pm 0.97\%$
MAML-MetaMix	$92.06 \pm 0.51\%$	$97.95 \pm 0.17\%$	$39.26 \pm 1.79\%$	$58.96 \pm 0.95\%$
MAML-ours	$\mathbf{92.89 \pm 0.46\%}$	$\mathbf{98.03 \pm 0.15\%}$	$\mathbf{39.89 \pm 1.73\%}$	$\mathbf{59.32 \pm 0.93\%}$
MetaSGD	$87.72 \pm 0.61\%$	$95.52 \pm 0.18\%$	$33.70 \pm 1.63\%$	$52.14 \pm 0.92\%$
MetaSGD-MetaMix	$93.59 \pm 0.45\%$	$98.24 \pm 0.16\%$	$40.06 \pm 1.76\%$	$60.19 \pm 0.96\%$
MetaSGD-ours	$\mathbf{93.93 \pm 0.40\%}$	$\mathbf{98.49 \pm 0.12\%}$	$\mathbf{40.22 \pm 1.78\%}$	$\mathbf{60.24 \pm 0.91\%}$
T-Net	$87.71 \pm 0.62\%$	$95.67 \pm 0.20\%$	$33.73 \pm 1.72\%$	$54.04 \pm 0.99\%$
T-Net-MetaMix	$93.27 \pm 0.46\%$	$98.09 \pm 0.15\%$	$38.33 \pm 1.73\%$	$59.13 \pm 0.99\%$
T-Net-ours	$\mathbf{93.54 \pm 0.49\%}$	$\mathbf{98.27 \pm 0.14\%}$	$\mathbf{38.38 \pm 1.77\%}$	$\mathbf{59.25 \pm 0.97\%}$



(a) Pose (b) Sinusoid (c) MiniImagenet (d) Omniglot

Figure 3: Ablation study.

different meta-learning backbones and compare them with our methods (MAML-Dropout+MAML-Bins) using 5 bins and dropout rate 0.1. Detailed settings are described in Appendix D.4.

We report our results in Table 4. Under a non-mutually-exclusive setting, our method significantly boosts gradient-based methods; even outperforms MetaMix, which proves that our methods have a better deconfounding ability. Besides, under the same setting, we investigate the influence of different hyperparameters, including different numbers of bins and dropout rates, on classification tasks. As shown in Figure 5a and Figure 5c in Appendix E.1, different hyperparameters improve the performance robustly. The ablation study on image classification is reported in Figure 3c and Figure 3d. Combining two implements of frontdoor adjustment has the best performance on these tasks. To show the effectiveness of our proposed methods, we compare pre-inner-update accuracy and meta-testing post-inner-update accuracy during meta-training under the Omniglot 20-way, 1-shot setting as shown in Table 5 in Appendix E.2. Additionally, we conduct the experiments on the mutually-exclusive setting of MiniImageNet in Appendix E.3.

6 Conclusion

In this paper, we rethink memorization overfitting from a causal perspective and construct a causal graph for gradient-based meta-learning. Under this causal graph, we identify the root cause of the memorization problem as a spurious correlation in meta-learning. Drawing upon our causal graph, we not only illustrate how existing methods solve the memorization problem but also propose a novel causal intervention principle to debias the spurious correlation. Two implementations of the proposed principle have demonstrated their effectiveness and compatibility in four benchmark datasets. More importantly, we believe that this causal perspective opens a new door to improving meta-learning.

REPRODUCIBILITY STATEMENT

We have conducted experiments under the setting in Section 5 and reported the results in Section 5 and Appendix E. We have provided dataset details in Appendix D and implementation details along with hyperparameter settings in Appendix D. We will release the code upon acceptance.

REFERENCES

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJGven05Y7.

Y Bengio, S Bengio, and J Cloutier. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pp. 969–vol. IEEE, 1991.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sebastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxWIgBFPS.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32:11698–11709, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkxStoC5F7.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ_UL-k0b.

Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11719–11727, 2019.

Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJE-4xW0W.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8747–8758, 2018.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Hae Beom Lee, Taewook Nam, Eunho Yang, and Sung Ju Hwang. Meta dropout: Learning to perturb latent features for generalization. In *International Conference on Learning Representations*, 2019.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2927–2936. PMLR, 2018.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

Eric J Martin, Valery R Polyakov, Xiang-Wei Zhu, Li Tian, Prasenjit Mukherjee, and Xin Liu. All-assay-max2 pqsar: activity predictions as accurate as four-concentration ic50s for 8558 novartis assays. *Journal of chemical information and modeling*, 59(10):4450–4459, 2019.

Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *ArXiv preprint*, abs/1910.01751, 2019. URL https://arxiv.org/abs/1910.01751.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. {BOIL}: Towards representation change for few-shot learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=umIdUL8rMH.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10860–10869, 2020.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgMkCEtPB.

Janarthanan Rajendran, Alexander Irpan, and Eric Jang. Meta-learning requires meta-augmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5705–5715. Curran Associates, Inc., 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4080–4090, 2017.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Hung-Yu Tseng, Yi-Wen Chen, Yi-Hsuan Tsai, Sifei Liu, Yen-Yu Lin, and Ming-Hsuan Yang. Regularizing meta-learning via gradient dropout. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pp. 75–82. IEEE, 2014.

Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020.

Huaxiu Yao, Long-Kai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, et al. Improving generalization in meta-learning via task augmentation. In *International Conference on Machine Learning*, pp. 11887–11897. PMLR, 2021.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2734–2746. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1cc8a8ea51cd0adddf5dab504a285915-Paper.pdf.

Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020.

Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pp. 7693–7702. PMLR, 2019.

A DETAILED PROOF

A.1 PROOF OF MERGED CAUSAL RELATION

We merge the causal relation $X_Q \to \hat{Y}'_Q \to \theta' \to \Phi \to \hat{Y}_Q$ to $X_Q \to \hat{Y}'_Q$. We have:

$$p(X_Q, \hat{Y}_Q, \hat{Y}'_Q, \Phi, \theta') = p(\hat{Y}_Q|X_Q, \Phi)p(\Phi|\theta')p(\theta'|\hat{Y}'_Q)p(\hat{Y}'_Q|X_Q)p(X_Q).$$

Given θ', then,

$$\begin{aligned} p(X_Q, \hat{Y}_Q, \Phi|\theta' = \theta^*) &= \sum_{\hat{Y}'_Q} p(\hat{Y}_Q|X_Q, \Phi)p(\Phi|\theta' = \theta^*)p(\hat{Y}'_Q|X_Q)p(X_Q) \\ &= p(\hat{Y}_Q|X_Q, \Phi)p(\Phi|\theta' = \theta^*)p(X_Q), \end{aligned} \tag{A.1}$$

whose graph is the same as Figure 1b.

A.2 THE CAUSAL EFFECT

Causal effect rule (Pearl et al., 2016) Given a causal graph G in which PA is a set of parent nodes of X, the causal effect of X on Y is given by

$$p(Y = y|do(X = x)) = \sum_z p(Y = y|X = x, PA = z)p(PA = z), \tag{A.2}$$

where z ranges over all the combinations of values that the variables in PA can take.

If a variable Z has no effect on Y, then we have

$$\begin{aligned} p(Y = y|do(X = x)) &= \sum_z p(Y = y|X = x, Z = z)p(Z = z) \\ &= \sum_z p(Y = y|X = x)p(Z = z) \\ &= p(Y = y|X = x). \end{aligned} \tag{A.3}$$

In this case, the correlation between X and Y is the causal effect of X on Y. However, if a variable Z has a effect on Y, then

$$\begin{aligned} p(Y = y|X = x) &= \sum_z p(Y = y|X = x, Z = z)p(Z = z|X = x) \\ &\neq \sum_z p(Y = y|X = x, Z = z)p(Z = z), \end{aligned} \tag{A.4}$$

so the correlation between X and Y is different from the causal effect. In this case, Z open the backdoor path of X and Y, which causes a spurious correlation.

In Figure 1b, there is no backdoor path between $\{S, \theta\}$ and $\{Q, \theta\}$, but Y open the backdoor path between $\{\theta', \theta\}$. Therefore, the causal effect of $\{\theta', S, Q\}$ on θ is

$$p(\theta|do(\theta', S, Q)) = p(\theta|do(\theta'), S, Q) \tag{A.5}$$

Frontdoor adjustment We apply frontdoor adjustment (Pearl et al., 2016) to calculate $p(\theta|do(\theta'), S, Q)$. Firstly, according to the causal graph Figure 1b, we have

$$p(\Phi|\theta', S) = P(\Phi|do(\theta'), S),$$

and,

$$p(\theta|do(\Phi), Q) = \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q)p(\theta'_i)$$

Then, the frontdoor adjustment for meta-learning is

$$\begin{aligned} p(\theta|do(\theta'), S, Q) &= \sum_{\Phi} p(\Phi|do(\theta'), S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) \\ &= \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) \end{aligned} \tag{A.6}$$

Complete causal graphs Complete causal graphs of two kinds of deconfounding methods mentioned in Section 2.3 and Section 2.4 are shown in Figure 4. Augmentation-based methods randomize the labels of query sets, *i.e.*, $Y' \leftarrow Q \rightarrow Y$. The frontdoor adjustment breaks the link $\theta' \rightarrow \Phi$. According to causal graphs, both these two kinds of methods solve the problem that Y is a confounder in meta-learning.

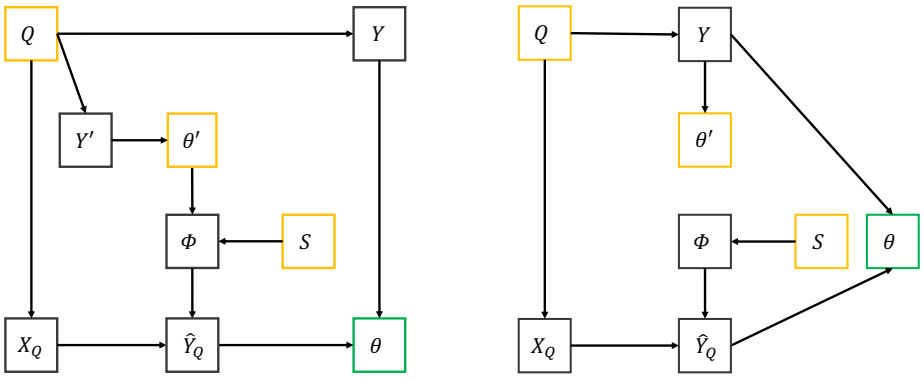


Figure 4: Complete causal graph of Figure 2b and Figure 2d. (a) The complete causal graph of augmentation-based methods. (b) The complete causal graph of the frontdoor adjustment.

B Detailed Algorithm

B.1 Pseudo-codes

We show the pseudo-codes of meta-training for MAML-Bins together with MAML-Dropout in Alg. 1.

B.2 Discussion of Our Two Methods

The two proposed methods sample stratification and deconfound in different manners: different stratums in MAML-Dropout are θ's dropping different parts of features, while different stratums in MAML-Bins are different combinations of existing features represented by θ'. They are complementary and mutually reinforcing, as evidenced in Figure 3. In general, MAML-Dropout tends to have more stratums than MAML-Bins, accounting for its better performance.

C Auxiliary Classification Task

To assign the images into different groups, we propose a novel method to train the feature extractor and groups the output of network with a standard clustering algorithm, kmeans. Thus, our method has two procedure: training stage and clustering stage.

Training stage. We train a feature extractor f_θ (parametrized by the network parameters θ) and the classifier $C(\cdot|W)$ (parametrized by the weight matrix $W \in \mathbb{R}^{d\times c}$) from scratch by minimizing

Algorithm 1 Meta-training Process of MAML-Dropout and MAML-Bins

Require: Task distribution $p(\mathcal{T})$; Learning rate α, β; A pretrained bin classifier C; Number of inner-loop steps t; Auxiliary classification loss weight λ; Dropout rate r. Randomly initialize parameter θ_0

while not coverage **do**

  Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{n}$

  for all $\mathcal{T}_i$ **do**

    Sample support set $s_i = \{(x_{i,j}^s . y_{i,j}^s)\}_{j=1}^{K_i^s}$ and query set $q_i = \{(x_{i,j}^q . y_{i,j}^q)\}_{j=1}^{K_i^q}$ from $\mathcal{T}_i$

    Classify the query set by C and match bin labels B to query set $q_i = \{(x_{i,j}^q . y_{i,j}^q), b_{i,j}^q\}_{j=1}^{K_i^q}$

    Compute the task-specific parameter $\phi_i = \phi_{i,t}$ on the support set using Eq.(3)

    Sample dropout masks $z_i \sim Bernoulli(r)$ for model f_{ϕ_i}

    Compute the output of the model f_{ϕ_i} with dropout masks z_i, *i.e.*,$O_i = f_{\phi_i, z_i}(X_i^q)$

    Compute the model prediction $\hat{Y}_i^q$ with the mean of O_i's column vectors and the bin prediction $\hat{B}_i^q$ with the mean of O_i's row vectors

    Compute the the loss as $\mathcal{L}(\hat{Y}_i^q, Y_i^q) + \lambda \mathcal{L}(\hat{B}_i^q, B_i^q)$

  end for

  Update $\theta_0 = \theta_0 - \frac{\beta}{n} \sum_{i=1}^{n} \nabla_{\theta_0} [\mathcal{L}(\hat{Y}_i^q, Y_i^q) + \lambda \mathcal{L}(\hat{B}_i^q, B_i^q)]$

end while

a standard cross-entropy classification loss L_{pred} using the training examples in the base classes $x_i \in X$. Here, we denote the dimension of the encoded feature as d and the number of output classes as c. The classifier $C(.|W)$ consists of a linear layer $W_\theta^T(x_i)$ followed by a softmax function σ.

Note that the training procedure in this model does not involve sampling mini-batches of classes and data points (episode) as in typical meta-learning algorithms.

Clustering stage. To assign the images into different groups, we fix the pre-trained network parameter θ in our feature extractor f_θ, and cluster the output of the network by a standard clustering algorithm, k-means. k-means takes the representation $f_\theta(x)$ as input, and clusters them into k distinct groups based on a geometric criterion. More precisely, it jointly learns a $d \times k$ centroid matrix C and the cluster assignments y_n of each image n by solving the following problem:

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|f_\theta(x_n) - C y_n\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1. \tag{A.1}$$

Solving this problem provides a set of optimal assignments $(y_n^*)_{n \leq N}$ and a centroid matrix C^*. These assignments are then used as pseudo-labels of bins; we make no use of the centroid matrix.

D Detailed Experimental Setup

D.1 Sinusoid Regression

To set up a toy sinusoid regression problem that is non-mutually-exclusive, we create data for each task in the following way: The data for each task is created in forms of $A \cdot \sin w \cdot x + b + \epsilon$, with $A \in [0.1, 5.0]$, $w \in [0.5, 2.0]$ and $b \in [0, 2\pi]$. At the test time, we expand the range of the tasks by randomly sampling the data-generating A uniformly from $[0.1, 5]$, w from $[0.5, 2.0]$, b from $[0, 2\pi]$ and use a one-hot vector for each (A, b), w as input to the network. The meta-training tasks are a proper subset of the meta-testing tasks. We set the number of bins to be 5, the dropout rate to be 0.3 and the weight of auxiliary task to be 1 in these tasks.

D.2 Pose Prediction

To preprocess the pose prediction tasks, we follow (Yin et al., 2020) to preprocess the pose tasks[1]. There are 50 and 15 categories in the meta-training and meta-testing, respectively, where each category contains 100 gray images in the size of 128×128.

[1] code link: https://github.com/google-research/google-research/tree/master/meta_learning_without_memorization/pose_data

Following Yin et al. (2020), in pose prediction task, the base model is comprised of a fixed encoder with three convolutional blocks and an adapted decoder with four convolutional blocks. Each convolutional block is composed of a convolutional layer, a batch normalization layer and a ReLU activation layer. We set the number of bins to be 5, the dropout rate to be 0.2 and the weight of auxiliary task to be 0.6 in these tasks.

D.3 Drug Activaty Prediction

This task comes from a public dose-response activity assay dataset from ChEMBL [2] and preprocessed by Martin et al. (2019). The training compounds in support sets and the testing compounds in query sets are separated by Martin et al. (2019) and the split of the meta-training, meta-validation and meta-testing tasks are as same as Yao et al. (2021).

The base model of drug activity prediction is a two-layer Multilayer Perceptron(MLP) neural network with 500 neurons in each layer. Each fully connected layer is followed by a batch normalization layer and leaky ReLU activation. In either meta-training or meta-testing, the number of inner-loop adaptation steps equals to 10. During meta-training, the task batch size, the outer-loop learning rate, the inner-loop learning rate are set to 8, 0.001 and 0.01. The meta-training process altogether runs for 50 epochs while 60 epochs using Dropout, each of which includes 500 iterations. Dropout rate is set to be 0.1. In order to prevent the influence of noise data, we use a query-set-mixup strategy as Yao et al. (2021), *i.e.*,we apply manifold mixup on query set for all experiments in this task.

D.4 Image Classification

In image classification, for non-mutually exclusive setting in 5-way miniImagenet, 64 meta-training classes are split to 5 sets, where 4 sets have 13 classes and the rest one has 12 classes. For each set, a fixed class label is assigned to each class within this set, which remains unchanged across different tasks. During meta-training, we randomly select one class from each set and take all the five selected classes to construct a task, which ensures that each class consistently has one label across tasks. In our experiments, we list the classes within each set as follows.

- **Set 1**: n07584110, n04243546, n03888605, n03017168, n04251144, n02108551, n02795169, n03400231, n03476684, n04435653, n02120079, n01910747, n03062245
- **Set 2**: n03347037, n04509417, n03854065, n02108089, n04067472, n04596742, n01558993, n04612504, n02966193, n07697537, n01843383, n03838899, n02113712
- **Set 3**: n04604644, n02105505, n02108915, n03924679, n01704323, n09246464, n04389033, n03337140, n06794110, n04258138, n02747177, n13054560, n04443257
- **Set 4**: n13133613, n01770081, n02606052, n02687172, n02101006, n03676483, n04296562, n02165456, n04515003, n01749939, n02111277, n02823428, n01532829
- **Set 5**: n02091831, n07747607, n03998194, n02089867, n02074367, n02457408, n04275548, n03220513, n03527444, n03908618, n03207743, n03047690

A similar process is applied to Omniglot, where 1200 meta-training classes are randomly split into 20 sets with 60 classes in each set. For all datasets, we utilize the classical convolutional neural network with 4 convolutional blocks as the base model (Finn et al., 2017; Snell et al., 2017). We set the number of bins to be 5, the dropout rate to be 0.1 and the weight of auxiliary task to be 0.2 in these tasks.

The image sizes of Omniglot and MiniImagenet are set to be $28 \times 28 \times 1$ and $84 \times 84 \times 3$, respectively.

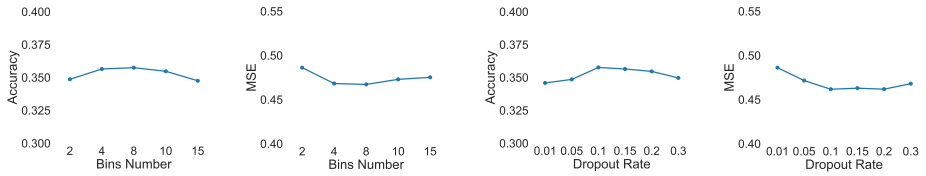
E Additional Experiment Results

E.1 Hyperparameter Sensitivity

The hyperparameters in our experiments are determined according to the performance on a hold-out set of meta-validation tasks. Besides, we analyze the influence of different numbers of bins for

[2] https://www.ebi.ac.uk/chembl

MAML-Bins and different dropout rate for MAML-Dropout. The results show the robustness of our methods against different hyperparameters.



(a) MiniImagenet (b) Sinusoid (c) MiniImagenet (d) Sinusoid

Figure 5: Hyperparameter analysis in (a - b) Bins Number (c - d) Dropout Rate.

E.2 Overfitting Analysis

We compare the shallow and deeper base model under the Omniglot 20-way 1-shot setting in Table 5. As for MAML, the memorization overfitting on the deep model is more serious, which really hurts the testing performance. Our methods solves the memorization problem in meta-knowledge achieves a better performance.

Table 5: Comparison between the shallow and deeper base model under the Omniglot 20-way 1-shot setting.

Methods	Meta-training Pre-update		Meta-testing Post-update	
	Shallow	Deep	Shallow	Deep
MAML	$14.38 \pm 0.40\%$	$98.59 \pm 0.05\%$	$87.40 \pm 0.59\%$	$8.82 \pm 0.42\%$
Ours	$5.46 \pm 0.38\%$	$5.07 \pm 0.41\%$	$92.11 \pm 0.39\%$	$84.37 \pm 0.59\%$

E.3 Results under Mutually-exclusive Setting

In Table 6, we report the results under the standard mutually-exclusive setting on MiniImagenet. Label shuffling is introduced to construct meta-training tasks under the mutually-exclusive setting, which significantly reduces the memorization overfitting. However, applying the proposed methods on this setting still achieves comparable and even better performance than original MAML, which further demonstrates the effectiveness of our proposed methods.

Table 6: Performance (Accuracy) of MiniImagenet under the mutually-exclusive setting.

Model	MiniImagenet	
	5-way 1-shot	5-way 5-shot
MAML	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
MAML-Bins	$49.18 \pm 1.70\%$	$63.85 \pm 0.97\%$
MAML-Dropout	$49.68 \pm 1.82\%$	$64.11 \pm 0.96\%$
MAML-Both	$\mathbf{50.06 \pm 1.76\%}$	$\mathbf{64.73 \pm 0.92\%}$

E.4 Results together with MetaMix

We apply our methods together with MetaMix to Omniglot, MiniImagenet and sinusoid regression. The results in the Table 7 and Table 8 show further and big improvement of the combination compared to using MetaMix only.

Table 7: Comparison with MetaMix on image classifications.

Model	Omniglot		MiniImagenet	
	20-way 1-shot	20-way 5-shot	5-way 1-shot	5-way 5-shot
MAML	$87.40 \pm 0.59\%$	$93.51 \pm 0.25\%$	$32.93 \pm 1.70\%$	$51.95 \pm 0.97\%$
MAML + MetaMix	$92.06 \pm 0.51\%$	$97.95 \pm 0.17\%$	$39.26 \pm 1.79\%$	$58.96 \pm 0.95\%$
MAML + ours	$92.89 \pm 0.46\%$	$98.03 \pm 0.15\%$	$39.89 \pm 1.73\%$	$59.32 \pm 0.93\%$
MAML + MetaMix + Ours	$\mathbf{93.02 \pm 0.68\%}$	$\mathbf{98.07 \pm 0.22\%}$	$\mathbf{39.92 \pm 1.77\%}$	$\mathbf{59.37 \pm 0.95\%}$

Table 8: Comparison with MetaMix on the sinusoid regression.

Model	5-shot	10-shot
MAML	0.59 ± 0.12	0.16 ± 0.06
MAML + MetaMix	0.47 ± 0.10	0.08 ± 0.02
MAML + ours	0.45 ± 0.08	0.06 ± 0.01
MAML + MetaMix + Ours	$\mathbf{0.44 \pm 0.09}$	$\mathbf{0.05 \pm 0.02}$