# The Role of Deconfounding in Meta-learning

**Yinjie Jiang** [*1] **Zhengyu Chen** [*1] **Kun Kuang** [1] **Luotian Yuan** [1]
**Xinhai Ye** [1] **Zhihua Wang** [2] **Fei Wu** [1 2 3] **Ying Wei** [4]

## 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.

[*]Equal contribution [1]Department of Computer Science and Technology, Zhejiang University, Hangzhou, China [2]Shanghai Institute for Advanced Study of Zhejiang University, Shanghai, China [3] Shanghai AI Laboratory, Zhejiang University, Shanghai, China [4]Department of Computer Science, City University of Hong Kong, Hong Kong, China. Correspondence to: Ying Wei <yingwei@cityu.edu.hk>, Kun Kuang <kunkuang@zju.edu.cn>.

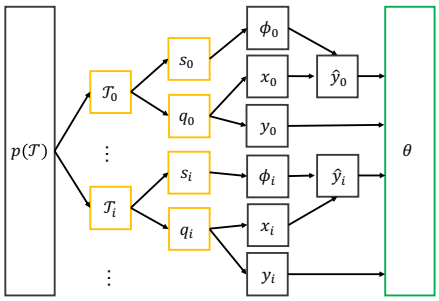
*Proceedings of the $39^{th}$ International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. 

## 1. Introduction

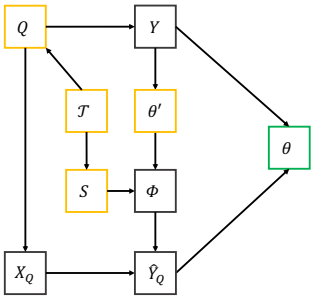
Recently, there has been renewed interest in meta-learning which empowers more human-like machines that suffice to learn a wide range of tasks with minimal supervision (Bengio et al., 1991; Thrun & Pratt, 2012; Finn et al., 2017; Raghu et al., 2020). While metric-based meta-learning algorithms (Vinyals 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 that should be only updated by the performance of task-specific models. Fortunately, the




(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 $\phi_i$ is updated by the support set $s_i$, then meta-knowledge $\theta$ is optimized on the likelihood of query sets $\{q_i\}_{i=1}^N$. (b) The causal graph of meta-learning, where $\theta'$ and $\theta$ are the initialization learned from last step and current step, respectively.

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.*, $\theta'$) and the task-specific model $\Phi$ 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 $\theta$ 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 $\theta$ a new task sampled from the same task distribution can be quickly learned with a resulting task-specific model $\phi$. 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, respectively.

Gradient-based meta-learning formulates learning of such an initialization $\theta$ as a bi-level optimization problem (see Figure 1a). During *inner-loop optimization*, the adapted model $\phi_i$ for the $i$-th task is initialized from $\theta$ and updated by its support set $s_i$. In outer-loop optimization, the initialization $\theta$ is optimized according to performances of adapted models on query sets, *i.e.*, by losses between label $y_{i,j}^q$ and prediction $\hat{y}_{i,j}^q$ of query samples. 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, i.e., $p(\phi|\theta, s)$, and the *outer-loop* optimizes the distribution of $\theta$, i.e., $p(\theta|D_{train})$. Consequently, the objective for *inner-loop* optimization (*a.k.a.,* task objective) is

$$\mathcal{L}(\phi_i) = \frac{1}{K^s} \sum_{j=1}^{K^s} \mathcal{L}(f_{\phi_i,\theta}(x_{i,j}^s), y_{i,j}^s), \tag{1}$$

and the objective for *outer-loop* optimization (*a.k.a.,* meta-objective) is

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{p(\phi_i|\theta,s_i)}[\frac{1}{K_i^q} \sum_{j=1}^{K_i^q} \mathcal{L}(f_{\phi_i,\theta}(x_{i,j}^q), y_{i,j}^q)].$$

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 $\theta$ 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

We would first like to introduce causality and causal graph, which are main theoretical tools we resort to. Next, 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 to 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. A causal graph (Pearl et al., 2016) addresses causality problems with a directed acyclic graph $G = \langle V, E\rangle$ where a node $V_i \in V$ denotes a variable and a directed edge $V_i \to V_j \in E$ denotes that the variable $V_i$ is a direct cause of $V_j$.

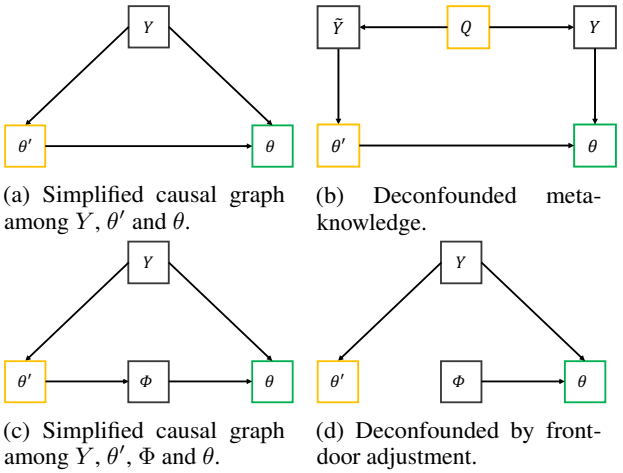
**Revisit of meta-learning: a causal view.** Given $N$ meta-training tasks during the meta-training phase, we define $s_i$ and $q_i$ to be the support set and the query set of task $i$, respectively. Then, in the meta-training data $D_{train}$, we have all support sets $\{s_i\}_{i=1}^N$ as $S$ and all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

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

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

**Deconfounded meta-learning.** In meta-learning, the meta-knowledge $\theta$ is learned by $p(\theta|\theta', S, Q)$ in each step, although the correlation between $\theta'$ and $\theta$ would be spurious since the causal path $\hat{Y}_Q \leftarrow \Phi \leftarrow \theta' \leftarrow Y \to \theta$ shown in Figure 1b demonstrates that $Y$ is a confounder of the path $\theta' \to \cdots \to \theta$ (see proof in A.2). Given $D_{train} = (S, Q)$, we simplify the causal graph with three nodes $\{Y, \theta', \theta\}$ as shown in Figure 2a via omitting the intermediate nodes. $Y$ opens the backdoor path from $\theta$ and $\theta'$. Unfortunately, the backdoor adjustment criterion (Pearl, 2009) is not applicable to break the causal relationship between $Y$ and $\theta'$ because the edges $\boldsymbol{Y \to \theta}$ and $\boldsymbol{Y \to \theta'}$ in the causal graph would change simultaneously since they play 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 the front-door criterion (Pearl, 2009). We will detail the two methods in Section 2.3 and Section 2.4.




(a) Simplified causal graph among $Y$, $\theta'$ and $\theta$.

(b) Deconfounded meta-knowledge.

(c) Simplified causal graph among $Y$, $\theta'$, $\Phi$ and $\theta$.

(d) Deconfounded by frontdoor adjustment.

Figure 2: Simplified causal graphs of meta-training and deconfounded methods.

### 2.3. Deconfounded Meta-knowledge

One possible solution to break the connection from $Y$ to $\theta'$ is to use different label mapping functions in different steps of meta-training as shown in Figure 2b. $\tilde{\boldsymbol{Y}} \leftarrow \boldsymbol{Q} \rightarrow \boldsymbol{Y}$ denotes two kinds of label representations of query sets. In this fork structure, $Y$ and $\tilde{Y}$ are independent, conditioned on query sets $Q$. The backdoor path from $Y$ to $\theta'$ 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 instantiations of the deconfounded meta-knowledge. Both methods randomize the labels of query sets to prevent the memorization. As spurious correlations are reduced, these methods achieve better performance than the vanilla 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 samples with manifold mixup (Verma et al., 2019) and channel shuffle. As a result, the meta-knowledge $\theta$ is equivalent to being optimized in even different label spaces in different steps.

In fact, even though these two methods reduce spurious correlations in the same manner, they perform quite differently on the same tasks. This phenomenon is attributed to the fact that conditioning on $Q$ makes only a partial of the spurious correlations blocked. The augmentation function sampled from a random space still confounds the model. To be specific, $Y'$ and $Y$ are not completely independent.

Another possible way to deconfounding the meta-knowledge is to constrain the meta-knowledge $\theta'$ so as to weaken the correlation between $Y$ and $\theta'$, which is adopted by Yin et al. (2020) via meta-regularization of parameters. However, such naive regularization which meantime limits $Y \rightarrow \theta$ weakens the effectiveness of fast adaptation in the inner-loop. As a result, regularizer-based methods struggle to address the *trade-off* between effectiveness and generalization. Besides, the weakened correlation between $\theta'$ and $\Phi$ still confounds the model.

### 2.4. Deconfounded Meta-learning Model

Taking the mediator $\Phi$ in the path from $\theta'$ to $\theta$ into consideration, we can alternatively simplify the causal path $\hat{Y}_Q \leftarrow \Phi \leftarrow \theta' \leftarrow Y \rightarrow \theta$ in Figure 1 with four nodes $\{Y, \theta', \Phi, \theta\}$ as shown in Figure 2c. Another renowned way of blocking the backdoor path from $\theta'$ to $\theta$ is to disconnect the path from meta-knowledge $\theta'$ to $\Phi$ via frontdoor adjustment. We propose a novel way to deconfound the meta-learning model in Figure 2d, and propose to calculate $p(\theta|do(\Phi), Q)$ instead of $p(\theta|\Phi, Q)$, which eliminates the confounder $Y$, *i.e.*, $p(\theta|do(\theta'), S, Q)$. This follows the "frontdoor adjustment", which is proved in Appendix A.2. The deconfounded meta-learning model is

$$\begin{aligned} p(\theta|do(\theta'), S, Q) &= \sum_{\Phi} p(\Phi|\theta', S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) \\ &= \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i), \end{aligned} \tag{2}$$

where $p(\Phi|\theta', S) = 1$ denotes the delta function. In Eq. (2), we stratify the confounded past meta-knowledge $\theta'$, *i.e.*, $\theta' = \{\theta'_i\}$, where $\theta'_i$ is a stratum of $\theta'$. $p(\theta|\Phi, \theta'_i, Q)$ denotes optimizing $\theta$ grouped by $\theta'_i$. Similarly, $\Phi$ is grouped in the same way. We propose two implementations of MAML to stratify $\theta'$ in Section 3. After frontdoor adjustment, we break the frontdoor path from $\theta'$ to $\Phi$. Therefore, the model would not memorize query sets of meta-training tasks.

## 3. Two Methods to Deconfounding MAML

### 3.1. MAML-Dropout

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

$$\begin{aligned} p(\theta|do(\Phi), Q) &= \int p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) d\theta'_i \\ &\approx \frac{1}{M} \sum_{i=1}^{M} p(\theta|\Phi, \theta'_i, Q) \\ &= \frac{1}{M} \sum_{i=1}^{M} p(\theta|\Phi, \theta', Q, z_i), \end{aligned} \tag{3}$$

where $M$ is the number of sampling, and $\theta'_i$ indicates a combination of $\theta'$ and $z_i$ which is a set of dropout variables sampled from the Bernoulli distribution. Note that $\theta'$ is independent with $z_i$.

Different from DropGrad (Tseng et al., 2020), we add dropout layers in the forward network only on query sets during meta-training. We adopt multi-step optimization in the inner-loop (Antoniou et al., 2019) to update almost all meta-knowledge in each training step, which avoids limitation of the model flexibility. In each training step, a batch of meta-training tasks are used to optimize the model, so empirically we sample different parts of $\theta'$ through the Monte Carlo method for different tasks in a batch to approximate results of Eq. (3). The meta-training objective of MAML-Dropout follows Eq. (4) as below.

$$\begin{aligned}
&\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}\frac{v_t}{K_i^q}\sum_{j=1}^{K_i^q}\mathcal{L}(f_{\phi_{i,t},z_{i,t}}(x_j^q), y_j^q),\\
&s.t.\ \phi_{i,t} = \begin{cases}\theta', & \text{if } t=0\\ \phi_{i,t-1} - \alpha\nabla_{\phi_{i,t-1}}\frac{1}{K_i^s}\sum_{j=1}^{K_i^s}\mathcal{L}(f_{\phi_{i,t-1}}(x^s), y^s)), & \text{otherwise}\end{cases}.
\end{aligned} \tag{4}$$

In Eq. (4), $N$ 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, the learned weights $\phi_{i,t}$ and random variable $z_{i,t}$ parameterize 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 the meta-knowledge in MAML extracts powerful features (Raghu et al., 2020), we propose another method to stratify $\theta'$ 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. Therefore, we have Eq. (2) as:

$$p(\theta|do(\Phi), Q) = \frac{1}{M}\sum_{M} p(\theta|\Phi, \theta'_m, Q). \tag{5}$$

We genertate $m$ feature groups by linear combination. In the $m$-th group, the feature $\text{feat}_m = f_{\theta'_m}(x)$, where $x$ is the input and $\theta'_m$ indicates the parameters that lead to this feature group. $\theta'_m$ consists parts of information of $\theta$, so that $\theta'_m$ can be considered as a stratification of $\theta'$.

We define an $N$-way $M$-bin task as an $N$-way classification with $M$ bins. We propose an auxiliary task as an $M$-class classification problem to assign each sample 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 index of a sample to be its auxiliary task label, *i.e.*, $b$. During the inner loop, the model learns the main task and the auxiliary bin classification task jointly, *i.e.*, the combination of a group of meta-knowledge. Denote the output of the model as $O = f_\phi(x)$, where $O \in \mathbb{R}^{M\times N}$. The prediction of the auxiliary bin classification task is $p(\hat{b}|x,\phi) = \frac{1}{N}\sum_{j=1}^{N} O_j^T$ and the prediction of of the main task 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, supposingthat $\lambda$ is the weight of bin classification loss, we have the outer-loop objective as the following Eq. (6).

$$\mathcal{L}(\theta) = \frac{1}{N_T}\sum_{i=1}^{N_T}\mathbb{E}_{p(\phi_i|\theta,s_i)}\{\frac{1}{K_i^q}\sum_{j=1}^{K_i^q}[\mathcal{L}(\hat{y}_j^q, y_j^q) + \lambda\mathcal{L}(\hat{b}_j^q, b_j^q)]\}, \tag{6}$$

As for a regression task, we split the range of the label space covered by all training samples into several intervals as bins and set the interval index of a sample as its auxiliary task label. The training process is like a "1-way $M$-bin classification", and its objective is the same as Eq. (6). The $M$-bin task reshapes the meta-knowledge $\theta'$ into $M$ stratifications, which effectively implements the frontdoor adjustment.

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 computational overhead. We combine MAML-Dropout and MAML-Bins in Algorithm 1 and discuss more details 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 the meta-knowledge and enable fast adaptation 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 query sets 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 the size of trainable parameters (Yin et al., 2020; Oh et al., 2021; Zintgraf et al., 2019), and discouraging the dissimilarity between different tasks (Jamal & Qi, 2019). The regularizer-based method, DropGrad (Tseng et al., 2020), applies dropout to support sets but not to query sets. The common positive sides of these methods is reduction of the *spurious correlation*, while

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

**Require:** Task distribution $p(\mathcal{T})$; Learning rate $\alpha$, $\beta$; A pretrained bin classifier $C$; Number of inner-loop steps $t$; Auxiliary classification loss weight $\lambda$; Dropout rate $r$. Randomly initialize parameter $\theta_0$
**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. (4)
    Sample dropout masks $z_i \sim Bernoulli(r)$ for model $f_{\phi_i}$
    Compute the output of the model $f_{\phi_i}$ 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**

they still use confounded past meta-knowledge that limits the flexibility of the 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. Though both methods empirically advance regularizer-based methods, they only partially block the backdoor path.

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 relations instead of correlations; meanwhile, the powerful representation ability of neural networks is beneficial to causal models for dealing with high-dimensional data. Most relevant to ours are (Bengio et al., 2020) and (Yue et al., 2020), both of which combine 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 several 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.

Table 1: Performance (MSE) comparison on the sinusoid regression problem.

| MODEL | 5-SHOT | 10-SHOT |
|---|---|---|
| IFSL | $0.59 \pm 0.15$ | $0.15 \pm 0.04$ |
| DROPGRAD | $0.57 \pm 0.15$ | $0.14 \pm 0.07$ |
| MR-MAML | $0.57 \pm 0.11$ | $0.10 \pm 0.02$ |
| META-AUG | $0.53 \pm 0.10$ | $0.10 \pm 0.02$ |
| ANIL | $0.54 \pm 0.10$ | $0.10 \pm 0.02$ |
| ANIL-METAMIX | $0.51 \pm 0.10$ | $0.08 \pm 0.02$ |
| **ANIL-OURS** | $\mathbf{0.49 \pm 0.10}$ | $\mathbf{0.08 \pm 0.02}$ |
| MAML | $0.59 \pm 0.12$ | $0.16 \pm 0.06$ |
| MAML-METAMIX | $0.47 \pm 0.10$ | $0.08 \pm 0.02$ |
| **MAML-OURS** | $\mathbf{0.45 \pm 0.08}$ | $\mathbf{0.06 \pm 0.01}$ |
| METASGD | $0.56 \pm 0.11$ | $0.14 \pm 0.04$ |
| METASGD-METAMIX | $0.46 \pm 0.10$ | $0.07 \pm 0.02$ |
| **METASGD-OURS** | $\mathbf{0.43 \pm 0.07}$ | $\mathbf{0.04 \pm 0.01}$ |
| T-NET | $0.54 \pm 0.11$ | $0.11 \pm 0.03$ |
| T-NET-METAMIX | $0.49 \pm 0.10$ | $0.08 \pm 0.02$ |
| **T-NET-OURS** | $\mathbf{0.47 \pm 0.09}$ | $\mathbf{0.07 \pm 0.02}$ |

### 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 $A \cdot \sin w \cdot x + b + \epsilon$ ,

Table 2: Performance comparison on 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-MetaMix* | 0.347 | 0.292 | 49 | 0.301 | 0.282 | 47 | 0.302 | 0.258 | 45 | 0.348 | 0.303 | 51 |
| 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-MetaMix* | 0.364 | 0.296 | 49 | 0.271 | 0.230 | 45 | 0.312 | 0.267 | 48 | 0.338 | 0.319 | 51 |
| MetaSGD-ours | 0.390 | 0.342 | 57 | 0.316 | 0.269 | 43 | 0.358 | 0.339 | 56 | 0.360 | 0.311 | 50 |

* : results from Yao et al.(2021)'s paper

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 a dropout rate of 0.3. 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 shows the comparison of various baselines combating memorization overfitting, where IFSL (Yue et al., 2020) only focuses on the meta-testing phase, so that it cannot improve the performance in a non-mutually-exclusive setting; MR-MAML (Yin et al., 2020) achieves a minor improvement, which accords with our analysis in Section 2.3. The error rate by Meta-Augmentation (Rajendran et al., 2020) is larger than MetaMix (Yao et al., 2021), as mapped labels in different steps generated by MetaMix are more random and independent. As expected, our methods bring a huge improvement when applied to different backbones, showing high compatibility.

We also evaluate our method separately, as shown in Figure 3b. We find that both MAML-Dropout and MAML-Bins contribute the performances and combining both 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 $[0.1, 0.3]$, give that an extremely low dropout rate stratifies $\theta'$ in similar ways while an extremely high dropout rate limits the generalization of the meta-knowledge. For the same reason, the optimal number of bins $M$ resides in the 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 against a specific target protein, whereas there are only a few labeled samples in the support set. We split the 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. We also evaluate the square of Pearson coefficient, denoted as $R^2$, between the predictions 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 activity values of compounds from biochemical experiments are noisy and the Pearson coefficient is more meaningful. For the same reason, MAML-Bins brings additional noise, so we only apply MAML-Dropout with a dropout rate of 0.1 in this experiment. We report the mean and median of $R^2$ values over all meta-testing assays, and we also report the numbers of assays of $R^2 > 0.3$ which shows the reliability in pharmacy. Results of our method are in Table 2. In four different groups, our method (MAML-Dropout) is capable of improving the performances by a large margin regarding 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 consisting of 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 a dropout rate of 0.2. Detailed settings are described in Appendix D.2.

In Table 3, we evaluate more algorithms in this experiment. We observe that it is difficult for regularizer-based methods to overcome memorization overfitting, especially under the 10-shot setting. If there are only a few samples in support

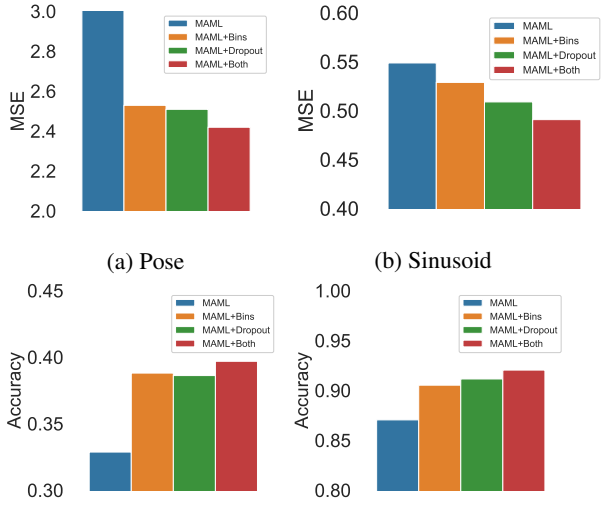
Table 3: Performance (MSE $\pm$ 95% confidence interval) comparison on pose prediction.

| MODEL | 10-SHOT | 15-SHOT |
|---|---|---|
| WEIGHT DECAY | $2.772 \pm 0.259$ | $2.307 \pm 0.226$ |
| CAVIA | $3.021 \pm 0.248$ | $2.397 \pm 0.191$ |
| META-DROPOUT | $3.236 \pm 0.257$ | $2.425 \pm 0.209$ |
| META-AUG | $2.553 \pm 0.265$ | $2.152 \pm 0.227$ |
| MR-MAML | $2.907 \pm 0.255$ | $2.276 \pm 0.169$ |
| IFSL | $3.186 \pm 0.256$ | $2.482 \pm 0.231$ |
| TAML | $2.785 \pm 0.261$ | $2.196 \pm 0.163$ |
| ANIL | $6.746 \pm 0.416$ | $6.513 \pm 0.384$ |
| ANIL-METAMIX | $6.354 \pm 0.393$ | $6.112 \pm 0.381$ |
| **ANIL-OURS** | $\mathbf{6.289 \pm 0.416}$ | $\mathbf{6.064 \pm 0.397}$ |
| MAML | $3.098 \pm 0.242$ | $2.413 \pm 0.177$ |
| MAML-METAMIX | $2.438 \pm 0.196$ | $2.003 \pm 0.147$ |
| **MAML-OURS** | $\mathbf{2.396 \pm 0.209}$ | $\mathbf{1.931 \pm 0.134}$ |
| METASGD | $2.803 \pm 0.239$ | $2.331 \pm 0.182$ |
| METASGD-METAMIX | $2.390 \pm 0.191$ | $1.952 \pm 0.154$ |
| **METASGD-OURS** | $\mathbf{2.369 \pm 0.204}$ | $\mathbf{1.926 \pm 0.112}$ |
| T-NET | $2.835 \pm 0.189$ | $2.609 \pm 0.213$ |
| T-NET-METAMIX | $2.563 \pm 0.201$ | $2.418 \pm 0.182$ |
| **T-NET-OURS** | $\mathbf{2.487 \pm 0.212}$ | $\mathbf{2.402 \pm 0.178}$ |

sets, the model is hard to adapt to a specific task and tends to memorize query sets in the meta training phase. The performances of the proposed method exceed MetaMix again, which highlights the deconfounding ability of our methods. In Figure 3a, either MAML-Bins or MAML-Dropout individuals still achieves a significant advancement compared with MAML itself.

### 5.4. Image Classification

We also study the memorization overfitting in a few-shot image classification problem with two benchmarks, *i.e.*, 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 again validates the power of deconfounding. We use a four-block convolutional network, which is the same as the model used in (Yao et al., 2021) which suffers from less meta-overfitting than the deeper neural network used in (Yin et al., 2020; Rajendran et al., 2020). We evaluate different meta-learning backbones and compare them with our methods (MAML-Dropout+MAML-Bins) using 5 bins and a dropout rate of 0.1. Detailed settings are described in Appendix D.4.




(a) Pose (b) Sinusoid

(c) MiniImagenet (d) Omniglot

Figure 3: Ablation studies of the proposed method on 4 different problems.

We report our results in Table 4. Under a non-mutually-exclusive setting, our method significantly boosts gradient-based methods; it even outperforms MetaMix, which proves that our methods are more capable of deconfounding than the baselines. Besides, under the same setting, we investigate the influence of different hyperparameters on classification tasks, including different numbers of bins and dropout rates. Figure 5a and Figure 5c in Appendix E.1 show that different hyperparameters improve the performance consistently, though the best hyperparameters further improve the performance. The ablation studies on these two image classification datasets are reported in Figure 3c and Figure 3d, respectively. Combining the two implementations of the frontdoor adjustment criterion arrives the best performance achieved so far. To further understand the superiority of our proposed methods, we compare the pre-inner-update accuracy and the 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 demon-

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** | $\mathbf{93.54 \pm 0.49\%}$ | $\mathbf{98.27 \pm 0.14\%}$ | $\mathbf{38.38 \pm 1.77\%}$ | $\mathbf{59.25 \pm 0.97\%}$ |

strated their effectiveness and compatibility in four benchmark datasets. More importantly, we believe that this causal perspective opens a new door to improving meta-learning.

## Acknowledgements

This work was supported in part by Key R&D Projects of the Ministry of Science and Technology (2020YFC0832500), Young Elite Scientists Sponsorship Program by CAST (2021QNRC001), National Natural Science Foundation of China (No. 62006207, No. 62037001), Project by Shanghai AI Laboratory (P22KS00111), the Fundamental Research Funds for the Central Universities (226-2022-00142), and Project 9229073 by RMGS of Research Grants Council (RGC), Hong Kong.

## References

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

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

Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxWIgBFPS.

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

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

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkxStoC5F7.

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

Jamal, M. A. and Qi, G.-J. 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.

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

Kurutach, T., Tamar, A., Yang, G., Russell, S., and Abbeel, P. Learning plannable representations with causal infogan. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8747–8758, 2018.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Lee, H. B., Nam, T., Yang, E., and Hwang, S. J. Meta dropout: Learning to perturb latent features for generalization. In *International Conference on Learning Representations*, 2019.

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

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Mahajan, D., Tan, C., and Sharma, A. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

Martin, E. J., Polyakov, V. R., Zhu, X.-W., Tian, L., Mukherjee, P., and Liu, X. 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.

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

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

Oh, J., Yoo, H., Kim, C., and Yun, S.-Y. {BOIL}: Towards representation change for few-shot learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=umIdUL8rMH.

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

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

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

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. 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.

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

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

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.

Tseng, H.-Y., Chen, Y.-W., Tsai, Y.-H., Liu, S., Lin, Y.-Y., and Yang, M.-H. Regularizing meta-learning via gradient dropout. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29:3630–3638, 2016.

Xiang, Y., Mottaghi, R., and Savarese, S. 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.

Yang, X., Zhang, H., and Cai, J. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020.

Yao, H., Huang, L.-K., Zhang, L., Wei, Y., Tian, L., Zou, J., Huang, J., et al. Improving generalization in meta-learning via task augmentation. In *International Conference on Machine Learning*, pp. 11887–11897. PMLR, 2021.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Yue, Z., Zhang, H., Sun, Q., and Hua, X.-S. Interventional few-shot learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (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.

Zhang, D., Zhang, H., Tang, J., Hua, X.-S., and Sun, Q. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. 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 $\theta'$, then,

$$\begin{aligned} p(X_Q, \hat{Y}_Q, \Phi|\theta' = \theta^*) &= \sum_{\hat{Y}'_Q} p(\hat{Y}_Q|X_Q, \Phi)p(\Phi|\theta' = \theta^*)p(\hat{Y}'_Q|X_Q)p(X_Q) \\ &= p(\hat{Y}_Q|X_Q, \Phi)p(\Phi|\theta' = \theta^*)p(X_Q), \end{aligned} \tag{7}$$

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

$$p(Y = y|do(X = x)) = \sum_z p(Y = y|X = x, PA = z)p(PA = z), \tag{8}$$

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

$$\begin{aligned} p(Y = y|do(X = x)) &= \sum_z p(Y = y|X = x, Z = z)p(Z = z) \\ &= \sum_z p(Y = y|X = x)p(Z = z) \\ &= p(Y = y|X = x). \end{aligned} \tag{9}$$

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

$$\begin{aligned} p(Y = y|X = x) &= \sum_z p(Y = y|X = x, Z = z)p(Z = z|X = x) \\ &\neq \sum_z p(Y = y|X = x, Z = z)p(Z = z), \end{aligned} \tag{10}$$

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 $\theta$ is

$$p(\theta|do(\theta', S, Q)) = p(\theta|do(\theta'), S, Q) \tag{11}$$

**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

$$\begin{aligned} p(\theta|do(\theta'), S, Q) &= \sum_{\Phi} p(\Phi|do(\theta'), S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) p(\theta|do(\Phi), Q) \\ &= \sum_{\Phi} p(\Phi|\theta', S) \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) \\ &= \sum_{\theta'_i} p(\theta|\Phi, \theta'_i, Q) p(\theta'_i) \end{aligned} \tag{12}$$

**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.

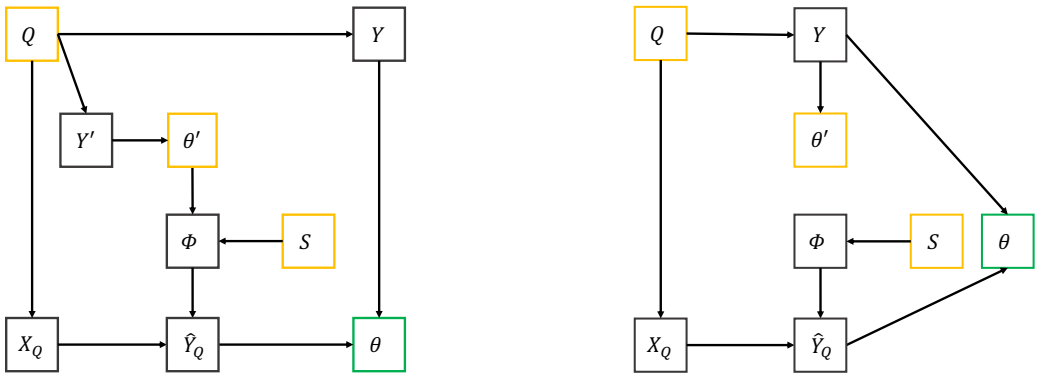



Figure 4: Complete causal graph of Figure 2b and Figure 2d. (a) The complete causal graph of augmentation-based methods. (b) The complete causal graph of the frontdoor adjustment.

## B. Detailed Algorithm

### B.1. Discussion of Our Two Methods

The two proposed methods sample stratification and deconfound in different manners: different stratums in MAML-Dropout are $\theta'$s dropping different parts of features, while different stratums in MAML-Bins are different combinations of existing features represented by $\theta'$. 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 $f_\theta$ (parametrized by the network parameters $\theta$) and the classifier $C(\cdot|W)$ (parametrized by the weight matrix $W \in \mathbb{R}^{d\times c}$) from scratch by minimizing a standard cross-entropy classification loss $L_{\text{pred}}$ 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 $\sigma$.

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 $\theta$ in our feature extractor $f_\theta$, 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:

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \|f_\theta(x_n) - C y_n\|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1. \tag{13}$$

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 $128 \times 128$.

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.

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

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

- **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

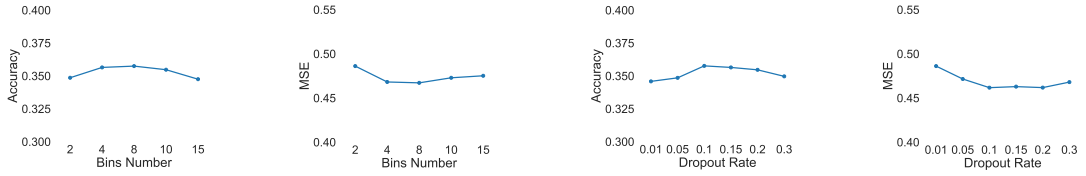
A similar process is applied to Omniglot, where 1200 meta-training classes are randomly split into 20 sets with 60 classes in each set. For all datasets, we utilize the classical convolutional neural network with 4 convolutional blocks as the base model (Finn et al., 2017; Snell et al., 2017). We set 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 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 | $0.59 \pm 0.12$ | $0.16 \pm 0.06$ |
| MAML + MetaMix | $0.47 \pm 0.10$ | $0.08 \pm 0.02$ |
| MAML + ours | $0.45 \pm 0.08$ | $0.06 \pm 0.01$ |
| MAML + MetaMix + Ours | $\mathbf{0.44 \pm 0.09}$ | $\mathbf{0.05 \pm 0.02}$ |