MetaCI: Meta-Learning for Causal Inference in a Heterogeneous Population

Ankit Sharma  
TCS Research, Delhi  
ankit.sharma16@tcs.com

Garima Gupta  
TCS Research, Delhi  
gupta.garima1@tcs.com

Ranjitha Prasad  
TCS Research, Delhi  
ranjitha.prasad@tcs.com

Arnab Chatterjee  
TCS Research, Delhi  
arnab.chatterjee4@tcs.com

Lovekesh Vig  
TCS Research, Delhi  
lovekesh.vig@tcs.com

Gautam Shroff  
TCS Research, Delhi  
gautam.shroff@tcs.com

Abstract

Performing inference on data obtained through observational studies is becoming extremely relevant due to the widespread availability of data in fields such as healthcare, education, retail, etc. Furthermore, this data is accrued from multiple homogeneous subgroups of a heterogeneous population, and hence, generalizing the inference mechanism over such data is essential. We propose the MetaCI framework with the goal of answering counterfactual questions in the context of causal inference (CI), where the factual observations are obtained from several homogeneous subgroups. While the CI network is designed to generalize from factual to counterfactual distribution in order to tackle covariate shift, MetaCI employs the meta-learning paradigm to tackle the shift in data distributions between training and test phase due to the presence of heterogeneity in the population, and due to drifts in the target distribution, also known as concept shift. We benchmark the performance of the MetaCI algorithm using the mean absolute percentage error over the average treatment effect as the metric, and demonstrate that meta initialization has significant gains compared to randomly initialized networks, and other methods.

1 Introduction

Learning causal relationships is the heart and soul of several domains such as healthcare, advertising, education, economics, etc. For instance, personalized and targeted treatment considering an individual's health indicators is crucial in healthcare [1, 2], targeted advertising campaign is essential to achieve higher profit margin in channel attribution [3, 4, 5]. Causal inference (CI) aims to infer unbiased causality effect of the treatment from observational data by factoring the impact of the confounding variables of patients/users. In the context of observational studies, confounding variables affect the treatment and outcome, and hence, disentangling the effect of these variables is the key to achieve treatment effectiveness. In this work, we tackle the fact that the study-population is heterogeneous, and hence, developing CI-based systems that generalize for new unseen subgroups in data is essential in order to provide better targeted interventions.
Classical approaches in CI estimate the average treatment effects from observational data by accounting for the selection bias using propensity scores, hence creating unbiased estimators of the averaged treatment effect (ATE) [6]. More recently, deep neural network based CI approaches have been proposed with different mechanisms to handle the bias. These include a latent variable modeling using VAEs [7], a GAN-based technique [8], a DNN-based Deep IV [9]. In [10] [11], the authors propose to view the causal inference problem as a covariate shift problem, and propose algorithms that balance between the factual and the counterfactual population.

Often, observational data is scarce, and the study-population is heterogeneous. Subgroup analysis is proposed in literature for coping with heterogeneity in the population [12] [13], especially in the context of establishing effect of the treatment for each subgroups [2]. Our goal is to design a deep neural network based causal inference model that is capable of adapting/generalizing to new subgroups in the input data that may not have been encountered during training. To achieve this goal, we use the novel ‘learning to learn’ paradigm, also known as the meta-learning framework. Unlike conventional deep neural networks that require large amounts of data for training, meta-learning or few-shot learning learns to learn from previous tasks by discovering the structure among tasks to enable fast learning of new tasks [14]. In this work, we employ the algorithmic framework for CI proposed in [10] [11], since it is a flexible framework in the context of meta-learning.

**Contributions**: We apply the meta optimization based technique known as Reptile on a well-known causal inference model [10]. A crucial design challenge is to define tasks, as in meta-learning context, appropriately for a given problem. Specifically, we define tasks based on features of the subgroups in such a way that tasks contain some commonality w.r.t to subgroups. In scenarios that have multiple substructures in the deep neural network model, we propose the ‘multi-Reptile’, which employs different learning rates for the parameters of the substructures.

As in [10], we assume that there is no hidden confounding. We demonstrate the results on two datasets – (a) synthetic dataset in the advertisement domain [3], and (b) semi-synthetic dataset based on the IHDP dataset [15]. We employ mean absolute percentage error (MAPE) defined on ATE as the metric, and demonstrate that our MetaCI framework counters the effect of heterogeneity in the input population and handles the change in target distributions during inference time, while the CI network counters the issue of covariate shift.

### 2 Preliminaries

In this section, we describe Reptile, an optimization based meta-learning paradigm, followed by description of the CI framework proposed in [10].
2.1 Meta-optimization preliminaries: Reptile

Reptile is an optimization based approach to meta-learning, where the model parameters are adapted for fast learning with a few examples. In [16], the authors state the optimization problem in this context for an initial set of parameters $\Psi$, a randomly sampled task $\omega$ with corresponding loss given by $L_{\omega}$, as follows

$$\Psi = \arg\min_{\Psi} \mathbb{E}_{\omega} \left[ L_{\omega}(U^{\omega}_{\epsilon}(\Psi)) \right],$$  \hspace{1cm} (1)$$

where $U^{\omega}_{\epsilon}(\cdot)$ is an update operator, and $L$ represents the stochastic gradient descent (SGD) epochs.

As an algorithm, Reptile involves repeatedly sampling task $\omega$, followed by learning the parameters using an update operator (e.g., SGD) on the data pertaining to $\omega$, and updating these parameters by learning on different tasks. The training phase of this framework provides a meta-initialization for the parameters $\Psi$ of the network, such that, for a new unseen task, network can be fine-tuned using this meta-initialization and a small amount of data from a new task. We employ the parallel version of reptile, where the solution for the optimization problem in (1) is given by

$$\Psi \leftarrow \Psi + \epsilon (\hat{\Psi} - \Psi),$$ \hspace{1cm} (2)$$

where $\epsilon$ is an adaptive learning rate, and $\hat{\Psi}$ is obtained after applying the update operator on the $\omega$-th task data. In this work, we consider the tasks pertaining to the causal inference where the goal is to learn a model for counterfactual inference. Hence, $U^{\omega}_{\epsilon}(\Psi)$ is a stochastic gradient descent operator which optimizes a cost function pertaining to counterfactual inference as given in [10]. We use the meta optimization framework to tackle both, the prior shift that occurs due to a drift in the feature distribution across tasks, and the concept shift that occurs due to a drift in probability distribution of the target variables [17]. In the sequel, we provide the basic setting of a causal inference problem, and describe the CI network which we use as the update operator, $U^{\omega}_{\epsilon}(\Psi)$.

2.2 Causal Inference preliminaries

In this subsection, we describe the problem of counterfactual inference in the meta-optimization framework. The CI network that we employ was proposed in [10, 11].

Let $T$ represent the set of treatments, $X^{\omega}$ be the set of contexts, and $Y^{\omega}$ be the set of possible outcomes w.r.t. the $\omega$-th task. We assume that the treatment is binary, that is $T \in \{0, 1\}$, where we assign treatment $t = 1$ as treated and $t = 0$ as control. Note that, for a given context $x^{\omega} \in X^{\omega}$, we observe one of the potential outcomes $y^{\omega} \in Y^{\omega}$, according to the treatment provided, i.e., if $t^{\omega} = 0$, we observe $y^{\omega} = Y^{\omega}_0$, and if $t^{\omega} = 1$, we observe $y^{\omega} = Y^{\omega}_1$, and accordingly we are interested in optimizing the ITE for the context in task $\omega$, $x^{\omega}$ is given by $ITE(x^{\omega}) = Y^{\omega}_1(x^{\omega}) - Y^{\omega}_0(x^{\omega})$. Furthermore, we are also interested in the the average treatment effect (ATE) averaged over all tasks and contexts, defined as $ATE = \mathbb{E}_{x^{\omega} \sim p(\omega)} \left[ \mathbb{E}_{x^{\omega} \sim p(x^{\omega})}ITE(x^{\omega}) \right]$.

In [10], the authors perform counterfactual inference by generalizing from the factual to counterfactual distribution. To this end, they learn a representation $\Phi^{\omega}$ and the function $h^{\omega}$, such that one term optimizes the prediction error w.r.t. the observed outcomes over the factual representation, the other term ensures that the distributions of treatment populations are similar or balanced for a given task $\omega$, thus tackling the issue of covariate shift [11]. Accordingly, the objective to minimize is

$$\mathcal{L}(\alpha^{\omega}, \gamma) = \frac{1}{N^{\omega}} \sum_{i=1}^{N^{\omega}} |h^{\omega}(\Phi^{\omega}(x^{\omega}_i, t_i) - y^{\omega}_i)| + \alpha^{\omega} \text{disc} \left( \hat{P}_{\Phi^{\omega}}, \hat{P}_{\Phi^{\omega}} \right) + \gamma \mathcal{R}(h^{\omega}),$$ \hspace{1cm} (3)$$

where $\alpha^{\omega}, \gamma > 0$ are hyperparameters that control the strength of the imbalance penalties, $\mathcal{R}(h^{\omega})$ is a model complexity term, $\hat{P}_{\Phi^{\omega}}$ represents the factual distribution, and $\hat{P}_{\Phi^{\omega}}$ represents the counterfactual distribution, respectively, and $\text{disc}(\cdot, \cdot)$ is the discrepancy measure as defined in [11].

3 MetaCI Model

In this section, we present the process of task creation, and describe the proposed MetaCI model.
3.1 Task creation

It is well known that a good meta-learning model should be trained for a diverse set of learning tasks and optimized based on the probability distribution of different tasks, including potentially unseen tasks. Defining task similarity is the key overarching challenge in meta learning. In the presence of heterogeneity in the population, we employ our knowledge regarding the features specific to subgroups, which are also the confounding variables in order to define tasks. We create tasks by combining a majority of samples from one subgroup, and a few samples from other subgroups in fixed proportions. Mathematically, using the joint distribution of the confounding variables, we ensure that we choose a subgroup that lies in a given region of the joint distribution, and mix it with samples from smaller disjoint regions of the same joint probability distribution, as depicted for a toy example in Fig. 1.

4 Proposed Model

In this section, we propose a novel MetaCI algorithm, where we combine a variant of the Reptile framework along with the causal inference framework [10]. As depicted in the neural network model in Fig. 2 we see that sampling of task and the update of weights using Multi-Reptile meta-learning algorithm occurs outside the CI block. The CI block constitutes the update operator in the context of meta-learning framework, and $L$ SGD epochs are used per meta-iteration. We term the meta-learning variant as Multi-Reptile, since it employs multiple adaptive learning rates for different subset of parameters of the update operator $U^L_{\omega}(\hat{W})$. Specifically, in the case of the CI network, we employ different learning rate for the representation and the hypotheses layers. The MetaCI algorithm is formally stated in Table. 1.

---

**Algorithm 1 MetaCI algorithm.**

1: procedure MetaCI(arguments)  
2: For all tasks, sample a test task $\omega \in \Omega_{te}$, and $\Omega_{tr}$ constitute the pool of train tasks.  
3: for $R$ iterations do  
4: Sample task $\omega \in \Omega_{tr}$.  
5: Compute the weights $\hat{W}_\phi$ and $\hat{W}_h$ using $U^L_{\omega}(W_\phi)$ and $U^L_{\omega}(W_h)$, respectively.  
6: Meta update weights of the representation layer:  
$$W_{\phi}^{r+1} = W_{\phi}^r + \epsilon_{\phi}(\hat{W}_\phi - W_{\phi})$$  
7: Meta update weights of the hypotheses layer:  
$$W_{h}^{r+1} = W_{h}^r + \epsilon_{h}(\hat{W}_h - W_{h})$$  

return $W_\phi$ and $W_h$.  

---
5 Experiments

In this section, we describe the datasets, the mechanism used for creating tasks for each dataset as described in Sec. 3.1 followed by the metrics we employ for evaluation, and finally the experimental results.

5.1 Datasets

We demonstrate the performance of the proposed algorithms on a synthetically generated advertisement dataset \cite{3} and the semi-synthetic IHDP dataset \cite{15}, for evaluation.

5.1.1 Synthetic advertisement dataset

We use a synthetic data generating process (DGP) to generate data relevant to the advertisement domain, as described in \cite{3}. We set the sample size \( N = 2000 \) and number of features \( p = 10 \). We generate features \( q_1, \ldots, q_p \sim N(0, 1) \), and the basis functions \( f_1(x), \ldots, f_{10}(x) \) as described in \cite{3}. We restrict the treatment \( T \) as being binary, and generate the treatment as 

\[
T | X = 1 \text{ if } \sum_{j=1}^{5} f_j(q_j) > 0,
\]

and 0 otherwise. Further, we generate the response as 

\[
Y | T, X \sim N(\sum_{j=1}^{5} f_j(q_j) + 5\eta T, \theta).
\]

We set \( \theta = 1 \) to generate data for demonstrating the effect of covariate shift, and set \( \theta \) as 1, 10 and 20 to generate data for demonstrating the effect of concept shift. Note that the features \( q_1, \ldots, q_5 \) have confounding effects on both the treatment and the outcome, and the rest of the features contribute to the noise in the model.

5.1.2 Semi-synthetic IHDP dataset

The Infant Health Development Program (IHDP) \cite{18} dataset consists of measurements of mother and children for studying the effect of specialist home visits on future cognitive test scores. The dataset comprises of 4302 infants having 25 features. Out of these, 8 are selected based on ACIC challenge (2017) to obtain context information \( X \). Specifically, these features form the basis of the meta-learning tasks obtained using the DGP \cite{15}.

5.2 Task creation for Reptile

Here we describe the process of task creation to demonstrate the performance of the MetaCI framework in the presence of covariate and concept shift, for the datasets provided in the previous section.

5.2.1 Covariate shift

**Tasks in synthetic dataset:** In order to appropriately provide tasks to the MetaCI framework in presence of covariate shift, we generate 2000 users distinguished based on the set of features, for number of tasks defined by cardinality of \( \Omega \). We consider these \( |\Omega| \) disjoint chunks, and mix it with samples from other chunks in the ratio 3 : 2, i.e., each task consists of 60% of samples from a given chunk, and 40% of samples in equal proportion from \( k \) other chunks. For every subgroup, \( T | X \) and \( Y | T, X \) is generated using a generating process specified in \cite{3}. In the single feature case, the data is split on the basis of the first feature which is one of the confounding variables. In the case of multiple confounding features, the data is split on the basis of the first two features which are confounding. We create tasks based on the joint distribution of the confounding features as outlined in Sec. 3.1.

**Tasks in IHDP dataset:** Here we create tasks for the MetaCI framework for the IHDP dataset, with an end goal of demonstrating the performance of the proposed algorithm in presence of covariate shift. We define tasks by dividing the entire population of infants, given as a finite number of contexts in the ACIC challenge dataset, 2017, into \( |\Omega| \) equal sized chunks. We create these chunks based on the joint distribution of multiple confounding features. Specifically, we consider mother’s age, child’s bilirubin level and mother’s place of birth. Each chunk is mixed with samples from other chunks in the ratio 3 : 2, i.e., each task dataset, \( X_\omega \), consists of 60% of samples from a given chunk, and 40% of samples in equal proportion from \( k \) other chunks. For each of the tasks, \( T \) and \( Y_\omega \) is generated synthetically using heteroskedastic, additive error DGP given in \cite{15}.

\[ \text{The simulated datasets will be available upon request from authors post publication of the paper.} \]
In both the above cases, the number of chunks used for mixing \((k)\) is an experimental variable and lies in range \([1, |\Omega| - 1]\).

### 5.2.2 Concept and covariate shift

**Tasks in synthetic and IHDP dataset scenario:** In order to demonstrate the performance of MetaCI in the presence of concept shift, we use two different generation processes which differ in generation of the response variable \(Y\). Accordingly, we describe two types of task creation as follows:

1. **Case 1:** Concept shift using 2 DGPs: Based on the confounding features of the datasets, we consider 4 chunks per DGP, and 3 chunks per DGP, in synthetic and IHDP datasets, respectively.

2. **Case 2:** Concept shift using 3 DGPs: We consider 3 chunks per DGP and 2 chunks per DGP, in synthetic and IHDP datasets, respectively.

In both the above cases, the chunks are mixed within and across groups by retaining 60\% of the samples of one chunk, and replacing the remaining 40\% with samples from other chunks, to create tasks. The mixed chunks contribute to generating the responses as dictated by the number of DGPs. Across DGPs, the parameters of the distribution which is used to sample \(Y|T, X\) is varied to demonstrate concept shift.

### 5.3 Metrics

In this subsection we describe the performance metrics used for evaluating proposed causal meta model. We use average treatment effect \((ATE_{\omega,r})\) for \(r\)-th test iteration and test task \(\omega\) as the performance metric, which is defined as

\[
ATE_{\omega,r} = \frac{\sum_{i=1}^{N_{\omega,t_1}} (y_{i,1} - \hat{y}_{i,0})}{2N_{\omega,t_1}} + \frac{\sum_{i=1}^{N_{\omega,t_0}} (\hat{y}_{i,1} - y_{i,0})}{2N_{\omega,t_0}},
\]

where \(y_{i,1} (y_{i,0})\) is the factual response to treatment \(t_i = 1 (t_i = 0)\) and \(\hat{y}_{i,1} (\hat{y}_{i,0})\) is its corresponding counterfactual response, \(N_{\omega,t_1}\) (\(N_{\omega,t_0}\)) are the number of samples in the task \(\omega\) that are offered treatment 1 (0). In order to eliminate any bias in the test set, we report the averaged \(ATE\) corresponding to the iteration that has the least averaged validation objective across test set of the meta-test tasks. In the following section, we report the mean absolute percentage error defined on the ground truth \(ATE\) \(ATE_G\), and the \(ATE\) obtained as above as follows:

\[
MAPE = \left|\frac{ATE_G - ATE}{ATE_G}\right|, \quad (5)
\]

i.e., lower values of MAPE indicate that the obtained \(ATE\) values are closer to the ground truth \(ATE\).

### 5.4 Experimental details and results

In this section, we report the experimental details and the results obtained. We split \(|\Omega| - 1\) train tasks and a test task as shown in Fig. 3. Every train task is divided in the ratio 1 : 1 corresponding to training and validation and test task is divided in the ratio 2 : 1 : 1 corresponding to training, validation and test sets. The MetaCI framework is trained for 1000 iterations by sampling a train task in each iteration. For each iteration \((r)\), weights \((W_r)\) of causal meta model are computed after \(L = 64\) epochs of mini-batch Stochastic Gradient Descent (SGD) over the batches of train set of train task. These weights \(W_r\) (where \(r = i\) during training of MetaCI) are then used to update the initial weights \(W_{0,i}\) present at the start of each iteration using reptile update Eq. (2).

We pick the best train task hyper-parameters (learning rate, dropout, \(\epsilon\)) correspond to the least value of validation loss function averaged across all iterations. We evaluate the performance on the test set of test task (refer Fig. 3) by tuning the meta causal models’ weights \((W_{0,j})\), where \(j\) is every 100\textsuperscript{th} iteration) for 64 epochs on the test task’s train set. Best hyper-parameters for test task is obtained in the same manner as discussed for training phase.

We repeat each experiment by considering each of \(|\Omega|\) tasks as meta test tasks, and report the averaged MAPE across test sets of each test task as in Fig. 3.
We consider two baselines for MetaCI. The first baseline is meta learning based reptile algorithm that uses the NN4 causal inference network. This baseline was presented in [10]. NN4 does not incorporate a representation layer $\Phi$, as compared to the CI neural network in [10], and hence it is a good baseline. The authors demonstrate the superiority of their proposed network as compared to NN4. In the tables that follow in the next section, we employ two variants of this baseline, namely MetaNN4, which uses meta-initialization, and RandomNN4, which uses random initialization, both along with NN4. By adopting NN4 along with meta learning, we verify that the gains obtained by using CI network as compared to NN4 is carried over when we use meta learning. In addition, we provide another baseline which consists of the CI network which is trained for large number of epochs over data from each task, but initialized using random initialization. This baseline helps us to gauge the performance of the CI network when the data is not provided in a meta learning fashion. We denote this baseline as CI in the tables that follow in the next section.

5.5 Results

We demonstrate the performance of MetaCI for varying number of tasks ($|\Omega|$), varying $k$, and $\epsilon$ using different settings for task creation, in the context of synthetic and semi-synthetic dataset discussed in previous section. We present the results pertaining to data that sees a covariate shift, and the combined effect of both, concept and covariate shift. Convergence is demonstrated in Fig. 4.

5.5.1 Covariate shift:

Varying number of subgroups ($|\Omega|$): We study the performance by measuring the MAPE for varying number of tasks ($|\Omega|$), varying $k$, and $\epsilon$ using different settings for task creation, in the context of synthetic and semi-synthetic dataset discussed in previous section. We present the results pertaining to data that sees a covariate shift, and the combined effect of both, concept and covariate shift. Convergence is demonstrated in Fig. 4.

Table 1: MAPE: Varying $|\Omega|$ ($k = |\Omega| - 1$), using single feature for task creation.

| $|\Omega|$ | Synthetic dataset | $N_{\omega} | \ CI_{\Omega} | \ MetaCI | \ MetaNN4 | \ RandomCI | \ RandomNN4 |
|---------|------------------|------------------|------------------|------------------|------------------|
| 4       | 2000             | 0.7305           | 0.3513           | 1.9400           | 1.5563           | 1.9991           |
| 7       | 2000             | 0.7088           | 0.2473           | 2.0993           | 1.3160           | 1.9348           |
| 9       | 2000             | 0.9487           | 0.4284           | 1.9995           | 1.2832           | 1.4622           |
| 11      | 2000             | 0.8036           | 0.3475           | 0.7929           | 1.2855           | 0.9831           |

Varying number of chunks used for mixing ($k$): We vary the number of mixing chunks $k$, for a fixed number of tasks $\Omega$, to study the effect of mixing on the performance as measured by MAPE. For $|\Omega| = 7$ and $|\Omega| = 6$, we see that varying $k$ leads to an improved value of ATE compared to the ground truth ATE in Table 4.
Table 2: MAPE: Varying $|\Omega| (N_\omega)$ using multiple features for task creation.

| $|\Omega|$ | IHDP | Synthetic dataset |
|---|---|---|
| $N_\omega$ | MetaCI | RandomCI | $N_\omega$ | MetaCI | RandomCI |
| 4 | 1144 | 0.5164 | 1.6896 | 4 | 2000 | 0.3276 | 1.1786 |
| 6 | 764 | 0.5112 | 1.8492 | 7 | 2000 | 0.3528 | 0.6976 |
| 8 | 498 | 1.7422 | 2.2367 | 9 | 2000 | 0.5762 | 0.6714 |

Figure 4: Comparison of validation objective (on test) across varying number of training epochs. (Left) IHDP dataset, (Right) Synthetic dataset.

**Varying meta learning rate $\epsilon$:** We demonstrate the relative performance of multi-reptile, where we vary the relative weights ($\epsilon$) assigned to the parameters of the representation layer ($W_\Phi$) vis-à-vis the weights assigned to the parameters of the hypotheses layer ($W_h$). Across several scenarios and datasets, as shown in Table 4, we observe that adopting a slower learning rate for the representation layer as compared to the hypotheses layer leads to $ATE$ very close to the ground truth $ATE$. Intuitively, the representation layer minimizes the discrepancy between distributions, which may vary slowly across tasks.

5.5.2 Concept and covariate shift

In this section, we present results for datasets in which we synthetically simulate concept and covariate shift at the same time. While covariate shift is inherent to the CI setting and arises due to confounding variables, concept shift arises due to the change in the probability distribution of the response variable conditioned on the input and treatment. In Table 5, we demonstrate the performance of the MetaCI algorithm when there are 2 and 3 DGP for generating the response as discussed in Sec. 5.2.2. Mean ($\mu_d$) and variance ($\sigma^2_d$) of $ATE$'s per DGP for both the datasets shown in Table 3, where $d = 1, 2, ...$. We observed that MetaCI converges faster as compared to RandomCI for both the datasets.

Table 3: MAPE: Varying $k$ using single feature ($|\Omega| = 7$) and multiple features ($|\Omega| = 6$ and $|\Omega| = 7$) for task creation.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Synthetic (single feature)</th>
<th>IHDP (multi features)</th>
<th>Synthetic (multi features)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MetaCI</td>
<td>RandomCI</td>
<td>MetaCI</td>
</tr>
<tr>
<td>3</td>
<td>0.4030</td>
<td>1.6603</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0.2490</td>
<td>1.4428</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td><strong>0.2473</strong></td>
<td><strong>1.3160</strong></td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4: Performance of the MetaCI framework for three scenarios, where speeds of relative weight adaptation of representation and hypotheses layer are varied.

| Scenario: $|\Omega| = 4, k = 3$ | $\epsilon_h = \epsilon_\phi$ | $\epsilon_h > \epsilon_\phi$ | $\epsilon_h < \epsilon_\phi$ |
|---|---|---|---|
| Multiple co-variate IHDP dataset | **0.4994** | 0.5164 | 0.7169 |
| Single co-variate synthetic dataset | **0.3230** | 0.3513 | 0.5966 |
| Multiple co-variate synthetic dataset | 0.3621 | **0.3276** | 0.5966 |
Table 5: Performance of MetaCI in case of covariate and concept shift using Synthetic and IHDP datasets.

<table>
<thead>
<tr>
<th># DGPs</th>
<th>((\mu_1, \sigma^2_1))</th>
<th>((\mu_2, \sigma^2_2))</th>
<th>((\mu_3, \sigma^2_3))</th>
<th>MetaCI</th>
<th>RandomCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>2</td>
<td>(0.4822, 0.0003)</td>
<td>(0.5356, 0.1739)</td>
<td>-</td>
<td><strong>0.4559</strong></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(0.5400, 0.0004)</td>
<td>(0.4733, 0.0337)</td>
<td>(0.7433, 0.4994)</td>
<td><strong>1.3153</strong></td>
</tr>
<tr>
<td>IHDP</td>
<td>2</td>
<td>(0.1515, 0.0002)</td>
<td>(0.9294, 0.0006)</td>
<td>-</td>
<td><strong>0.9419</strong></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(0.1521, 0.0007)</td>
<td>(0.9000, 0.0006)</td>
<td>(1.0288, 0.0080)</td>
<td><strong>1.6135</strong></td>
</tr>
</tbody>
</table>

6 Conclusions

In this work, we demonstrate the efficacy of the meta learning based reptile framework in a causal inference setting for a heterogeneous population. We showed that the meta learning approach is a modern approach that could replace the classical subgroup analysis, where these subgroups can be translated as tasks in the meta learning framework. We provided a novel approach to create tasks based on the confounding features, and showed that it is possible to obtain a good meta initialisation which leads to significant improvement in ATE on the unseen data. We also showed that the MetaCI framework adapts its parameters in the presence of both covariate and concept shift in the dataset, and outperforms the baselines by large margins. We allude to specific details regarding training meta learning based deep neural network models, which by itself is a contribution to current literature.

References


