ROTOGBML: TOWARDS OUT-OF-DISTRIBUTION GENERALIZATION FOR GRADIENT-BASED META-LEARNING

Anonymous authors
Paper under double-blind review

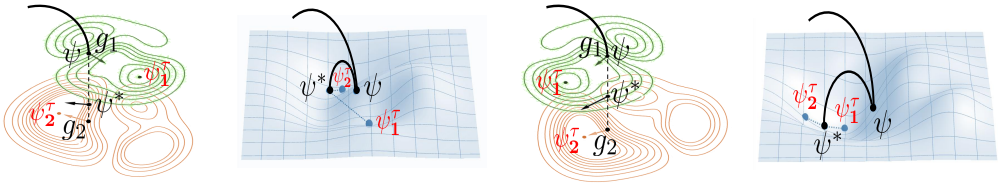
ABSTRACT

Gradient-based meta-learning (GBML) algorithms are able to fast adapt to new tasks by transferring the learned meta-knowledge, while assuming that all tasks come from the same distribution (in-distribution, ID). However, in the real world, they often suffer from an out-of-distribution (OOD) generalization problem, where tasks come from different distributions. OOD exacerbates inconsistencies in magnitudes and directions of task gradients, which brings challenges for GBML to optimize the meta-knowledge by minimizing the sum of task gradients in each minibatch. To address this problem, we propose RotoGBML, a novel approach to homogenize OOD task gradients. RotoGBML uses reweighted vectors to dynamically balance diverse magnitudes to a common scale and uses rotation matrixes to rotate conflicting directions close to each other. To reduce overhead, we homogenize gradients with the features rather than the network parameters. On this basis, to avoid the intervention of non-causal features (*e.g.*, backgrounds), we also propose an invariant self-information (ISI) module to extract invariant causal features (*e.g.*, the outlines of objects). Finally, task gradients are homogenized based on these invariant causal features. Experiments show that RotoGBML outperforms other state-of-the-art methods on various few-shot image classification benchmarks.

1 INTRODUCTION

Deep learning has achieved great success in many real-world applications such as visual recognition (Krizhevsky et al., 2012; Wang et al., 2020; Srinivas et al., 2021) and natural language processing (Vaswani et al., 2017; Devlin et al., 2018; Wu et al., 2016). However, deep learning relies heavily on large-scale training data, showing the limitation of not being able to effectively generalize to small data regimes. To overcome this limitation, researchers have explored and developed a variety of meta-learning algorithms (Vanschoren, 2019; Wang, 2021; Raghu et al., 2020; Thrun & Pratt, 2012; Schmidhuber, 1987), whose goal is to extract the meta-knowledge over the distribution of tasks rather than instances, and hence compensate for the lack of training data. Among the two dominant strands of meta-learning algorithms, we prefer gradient-based (Finn et al., 2017; Yoon et al., 2018; Yin et al., 2020; Lee et al., 2019a) over metric-based (Snell et al., 2017; Vinyals et al., 2016) for their flexibility and effectiveness. Unfortunately, most of these researches have a restrictive assumption that each task comes from the same distribution (in-distribution, ID). However, distribution shifts among tasks are usually inevitable in real-world scenarios (Shen et al., 2021; Amrith et al., 2021).

In this paper, we consider a realistic scenario, where each minibatch is constructed of tasks from different distributions or datasets (out-of-distribution, OOD). Surprisingly, through repeated experiments, we found that the OOD tasks seriously affect the performance of GBML, *e.g.*, the performance of MAML drops from 75.75% to 54.29% with CUB dataset under the 5-way 5-shot setting. Intuitively, we explain the possible reason from the optimization objective of the meta-knowledge. Specifically, GBML algorithms learn the meta-knowledge by minimizing the sum of gradients for each minibatch of tasks (see equation (1)). If task gradients have a significant inconsistency, it may cause the learned meta-knowledge to be dominated by certain tasks with large gradient values and fail to generalize to new tasks, affecting the performance of GBML algorithms. The OOD generalization problem exacerbates the phenomenon, where task gradients are inconsistent. To demonstrate the impact of the OOD problem on task gradients in each minibatch, we next give an illustrative example.



(a) Task space of GBML (b) Meta space of GBML (c) Task space of Ours (d) Meta space of Ours

Figure 1: Visualization OOD task-gradient magnitudes and directions for GBML and RotoGBML (Ours) in task and meta spaces. $\psi = (\theta, \phi)$ and ψ^* are initial and updated meta parameters, respectively. g_1 and g_2 are task gradients with task parameters ψ_1^τ and ψ_2^τ in outer loop, respectively.

We take two different distributions to evaluate the impact in Figure 1. The optimization process of meta-knowledge in GBML has two loops: an inner-loop at task space (Figure 1 (a) and (c)) and an outer loop at meta space (Figure 1 (b) and (d)). We randomly sample two OOD tasks (task1 and task2) from the two different distributions and represent them using green and gray contour lines, respectively. First, in the task space, each task learns a task-specific parameter ψ_1^τ or ψ_2^τ using the same meta-knowledge ψ. Then, in the meta space, the losses of two OOD tasks are calculated using ψ_1^τ and ψ_2^τ and are summed in turn to optimize the meta-knowledge from ψ to ψ^*. From Figure 1 (a), it clearly shows that in the whole optimization process, task2 has a large gradient value (the length of $\|g_2\| > \|g_1\|$) and the large gradient dominates the learning process, *i.e.*, the updated meta-knowledge ψ^* is close to task2 in Figure 1 (b). This causes the learning of meta-knowledge to be dominated by task2 and ignore the existence of task1, and eventually GBML cannot fast adapt to new tasks using the learning meta-knowledge. When the gradient directions of OOD tasks are conflicting, the meta-knowledge optimized by summing two task gradients may counteract each other.

In this paper, to solve inconsistencies in task-gradient magnitudes and directions, we propose a simple yet effective framework, RotoGBML, to simultaneously homogenize magnitudes and directions and boost the learning of meta-knowledge in GBML. Specifically, (1) RotoGBML solves the gradient magnitudes by dynamically reweighting task gradients at each step of the learning process, while encouraging the learning of ignored tasks. (2) Instead of directly modifying gradient directions, RotoGBML smoothly rotates each task space, seamlessly aligning gradient directions in the long run (see Figure 1 (c)). (3) To reduce overhead, we use the features instead of network parameters to homogenize task gradients, see Section 3.2 for details. Furthermore, we also propose an invariant self-information (ISI) module to extract invariant causal features (*e.g.*, object outline), which are used for homogenization. The introduction of ISI is mainly because of the fact that the features learned by neural networks are inevitably interfered with by some non-causal features (*e.g.*, image backgrounds), which in turn affects the homogeneity of task gradients. We theoretically demonstrate that OOD affects meta-knowledge learning. The main contributions could be briefly summarized as follows:

- We consider a real-world OOD scenario and propose a general RotoGBML algorithm to solve the OOD generalization problem. RotoGBML helps GBML algorithms learn good meta-knowledge by homogenizing task-gradient magnitudes and directions.
- To reduce memory, we homogenize task gradients at a feature level. Also, we design an invariant self-information module to extract the invariant causal features. Homogenizing gradients using these features provides further guarantees for learning robust meta-knowledge.
- We theoretically evaluate our motivation and experimentally demonstrate the effectiveness of our RotoGBML algorithm in various few-shot image classification benchmarks.

2 PRELIMINARIES

Task formulation in GBML. In this paper, a supervised learning setting is considered, where each data point is denoted by (x, y) with $x \in X$ being the input and $y \in Y$ being its corresponding label. GBML assumes N training tasks $\{\mathcal{T}_i\}_{i=1}^N \sim \mathcal{T}_{tr}$ in each minibatch to be sampled, and an arbitrary testing task $\mathcal{T}_i \sim \mathcal{T}_{te}$ is picked. Formally, each n-way k-shot task $\mathcal{T}_i$ consists of a support set $\mathcal{S}_i = (x_i^s, y_i^s)$ and a query set $\mathcal{Q}_i = (x_i^q, y_i^q)$. $x_i^s \in \mathbb{R}^{n_s \times d}$, $x_i^q \in \mathbb{R}^{n_q \times d}$, $y_i^s \in \mathbb{R}^{n_s \times n}$ and $y_i^q \in \mathbb{R}^{n_q \times n}$, where $d = c \times h \times w$ indicates the image size, n_s and n_q denote the number of support and query examples, respectively. The label spaces of training, validation and testing tasks are different, *i.e.*, $Y_{tr} \cap Y_{val} \cap Y_{te} = \emptyset$. Most GBML algorithms have a strict assumption for tasks as:

Assumption 2.1. *In-distribution (ID): each task is randomly sampled from the same distribution* $\mathbb{P}(\mathcal{T})$*, i.e.,* $\mathcal{T}_i \sim \mathbb{P}(\mathcal{T})$ *comes from the task set* $\{\mathcal{T}_{tr}, \mathcal{T}_{val}, \mathcal{T}_{te}\}$ *with* $\forall i$*.*

The assumption indicates a highly restrictive setting due to the following reasons: (1) It is contradictory that tasks with disjoint classes come from the same distribution. (2) The data collection process is susceptible to some unobservable variables, which can cause distribution shifts even from the same dataset. Improving the generalization of GBML algorithms in the real world is very important.

Task optimization in GBML. GBML aims to learn a good meta-knowledge and fast adapts to new tasks using two optimization loops (inner loop and outer loop). Generally, the network architectures of GBML contain a feature encoder f_θ parameterized by θ, which is used to extracted features $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$, and a classifier c_ϕ parameterized by ϕ, which is used to output predict labels $c_\phi : \mathbb{R}^m \to \mathbb{R}^n$. In the inner loop, the model is grounded to an initialization (or meta-knowledge), *i.e.*, (ϕ, θ), which is adapted to the i-th task in a few gradient steps τ (typically $1 \sim 10$ steps) by its support set $\mathcal{S}_i$. In the outer loop, the performance of the adapted model, *i.e.* $(\phi_i^\tau, \theta_i^\tau)$, is measured on the query set $\mathcal{Q}_i$, and in turn used to optimize the initialization from (ϕ, θ) to (ϕ^*, θ^*). Let $\mathcal{L}$ denotes the loss function and the above interleaved process is formulated as a bi-level optimization problem,

$$(\phi^*, \theta^*) := \min_{(\phi,\theta)} \frac{1}{N} \sum\nolimits_{i=1}^{N} \mathbb{E}_{(x_i^q, y_i^q)\sim\mathcal{Q}_i} \mathcal{L}_i(\phi_i^\tau(\theta_i^\tau(x_i^q)), y_i^q), \tag{1}$$

$$\text{s.t. } (\phi_i^\tau, \theta_i^\tau) \leftarrow \min_{(\phi_i^0, \theta_i^0)} \mathbb{E}_{(x_i^s, y_i^s)\sim\mathcal{S}_i} \mathcal{L}_i(\phi_i^0(\theta_i^0(x_i^s)), y_i^s), \tag{2}$$

where equations (1) and (2) are the outer-loop and inner-loop, respectively. $(\phi_i^0, \theta_i^0) = (\phi, \theta)$ is the initial parameters for the i-th task. The learning of meta-knowledge uses the summed and averaged gradients over all tasks in current batch. We introduce detailed derivations for GBML in Appendix B.

3 METHODOLOGY

To improve the generalization ability of GBML algorithms in real-world scenarios, in this paper, we consider an out-of-distribution (OOD) setting, *i.e.*, all tasks randomly sampled from a distribution set $\{\mathbb{P}_i(\mathcal{T})\}_{i=1}^N$. A new assumption on the OOD task distributions is proposed as follows:

Assumption 3.1. *Out-of-distribution (OOD): each task comes from the different distributions* $\{\mathbb{P}_i(\mathcal{T})\}_{i=1}^N$*, i.e.,* $\mathcal{T}_i \sim \mathbb{P}_i(\mathcal{T})$ *and* $\mathcal{T}_j \sim \mathbb{P}_j(\mathcal{T})$ *with* $\forall i, j$*,* $i \neq j$ *and* $\mathbb{P}_i(\mathcal{T}) \neq \mathbb{P}_j(\mathcal{T})$*.*

To solve the problem that OOD exacerbates the inconsistencies of task-gradient magnitudes and directions, we introduce RotoGBML, a novel algorithm that consists of two building blocks: (1) *Reweighting OOD task-gradient magnitudes*. A reweighted vector set $\mathbf{w}_\omega = \{\omega_i\}_{i=1}^N$ parameterized by ω is used to normalize the magnitudes to a common scale (see Section 3.1); (2) *Rotating OOD task-gradient directions*. A rotation matrix set $R_\gamma = \{\gamma_i\}_{i=1}^N$ parameterized by γ is used to rotate the directions close to each other (see Section 3.2). The two blocks complement each other and facilitate meta-knowledge to learn common information of all tasks in each minibatch, thereby fast adapting to new tasks with only a few samples. The optimization of RotoGBML is re-defined as follows:

$$(\phi^*, \theta^*) := \min_{(\phi,\theta)} \frac{1}{N} \sum\nolimits_{i=1}^{N} \mathbb{E}_{(x_i^q, y_i^q)\sim\mathcal{Q}_i} \omega_i \mathcal{L}_i(\phi_i^\tau(\gamma_i \theta_i^\tau(x_i^q)), y_i^q), \tag{3}$$

$$\text{s.t. } (\phi_i^\tau, \theta_i^\tau) \leftarrow \min_{(\phi_i^0, \theta_i^0)} \mathbb{E}_{(x_i^s, y_i^s)\sim\mathcal{S}_i} \mathcal{L}_i(\phi_i^0(\theta_i^0(x_i^s)), y_i^s). \tag{4}$$

We use red to annotate the differences between the generic GBML and RotoGBML. It clearly shows that a set of corresponding ω_i and γ_i for each task at each batch is used to homogenize gradients in the **outer loop** optimization process. And, if task-gradient magnitudes and directions are consistent, the equation (3) degenerates into the equation (1). It is worth mentioning that our RotoGBML algorithm is a model-agnostic method that can be equipped with arbitrary GBML algorithms (see Section 6).

3.1 REWEIGHTING OOD TASK-GRADIENT MAGNITUDES

How to determine the value of $\mathbf{w}_\omega$? This is a key challenge for reweighting OOD task-gradient magnitudes. Most previous works on multi-task learning use static approaches, such as hyperparameters or prior assignment (Kendall et al., 2018; Crawshaw, 2020). However, these methods are difficult

to adapt to the learning process due to the use of fixed values $\mathbf{w}_\omega = \{\omega_i\}_{i=1}^N$ for each task. In Table 3, the experiments of static reweighted vector evaluate the phenomenon. Instead, we dynamically adjust ω_i to adapt task distribution shifts by using an optimization strategy over the training iterative process.

Specifically, we initialize $\{\omega_i = 1 | i \in N\}$ for each task in each batch $\mathcal{T}_b = \{\mathcal{T}_i | i \in N\}$, which aims to treat each task equally at the beginning. Then, the reweighted vector set is dynamically adjusted by using average gradient norms for the current batch during training. The optimization objective is as:

$$\omega^* := \min_{\omega} \sum\nolimits_{i=1}^{N} \mathbb{E}_{(x_i^q, y_i^q) \sim \mathcal{Q}_i} \mathcal{L}_{\omega_i}(\theta_i^\tau, x_i^q, y_i^q), \;\; \text{s.t. } \mathcal{L}_{\omega_i} = \| g_{wi} - \bar{g}_{wb} \times [I_i]^\beta \|_1, \tag{5}$$

where $\| \cdot \|_1$ is the ℓ_1-norm. $g_{wi} = \| \nabla_\theta \omega_i \mathcal{L}_i(\theta_i^\tau(x_i^q), y_i^q) \|_2$ is the ℓ_2-norm of the gradients for each reweighted task loss $\omega_i \mathcal{L}_i$ in outer loop. $\bar{g}_{wb} = \frac{1}{N} \sum_{i=1}^N g_{wi}$ is the average gradient for $\mathcal{T}_b$. To save compute costs, we only use the gradients of feature encoder θ since the feature knowledge can be reused (Raghu et al., 2020). I_i is inverse learning rate to balance each task gradient as follows:

$$I_i = \hat{\mathcal{L}}_i / \sum\nolimits_{i=1}^{N} \hat{\mathcal{L}}_i, \quad \text{s.t. } \hat{\mathcal{L}}_i = \mathcal{L}_i / \mathcal{L}_i^0, \tag{6}$$

where $\mathcal{L}_i$ is the cross-entropy loss for query set in outer loop and $\mathcal{L}_i^0$ is the initial loss to be determineed by using a theoretical initial, *i.e.*, $\mathcal{L}_i^0 = log(n)$, n is classes. Concretely, the higher the value of I_i, the higher the gradient magnitudes should be for i-th task to encourage the task to train more quickly.

In equation (5), an additional hyperparameter β is introduced to pull tasks back to a common learning rate. When tasks are very different in their distributions, a higher β should be used to enforce stronger training rate balancing. Conversely, a lower β is appropriate for similar tasks. Note that $\beta = 0$ tries to pin the backpropagated gradients of each task to be equal at network parameters. The dynamically optimized ω_i can be integrated into the learning process to homogenize task-gradient magnitudes. In the next section, we describe how to resolve conflicts of task directions in each batch.

3.2 Rotating OOD Task-Gradient Directions

How to define and learn the matrix of R_γ? This is a key challege for rotating OOD task-gradient directions. (1) **Definition.** Motivated by previous work (Javaloy & Valera, 2022), we initialize $R_\gamma \in SO(M)$, where $SO(M)$ is special orthogonal group to denote the set of all rotation matrix γ_i with size M (the network parameters). Due to the large size of M, to reduce overhead, we adopt the usual practice and focus on feature-level task gradients $\nabla_{\hat{z}_i^q} \mathcal{L}_i$ (rather than $\nabla_\theta \mathcal{L}_i$). This is reasonable due to chain rule $\nabla_\theta \mathcal{L}_i = \nabla_\theta \hat{z}_i^q \cdot \nabla_{\hat{z}_i^q} \mathcal{L}_i$, where $\hat{z}_i^q = \gamma_i z_i^q$, $z_i^q = \theta_i^\tau(x_i^q)$. Finally, $R_\gamma \in SO(m)$ is used, where m is the size of feature dimensions. (2) **Learning.** R_γ aims to reduce the direction conflict of the task gradients in each batch by rotating the feature space. We optimize γ_i to make task spaces closer to each other and the objective is to maximize the cosine similarity or to minimize

$$\gamma^* := \min_{\gamma} \sum\nolimits_{i=1}^{N} \mathbb{E}_{(x_i^q, y_i^q) \sim \mathcal{Q}_i} \mathcal{L}_{\gamma_i}(\theta_i^\tau, x_i^q, y_i^q), \quad \text{s.t. } \mathcal{L}_{\gamma_i} = - \left\langle \gamma_i g_i, \bar{g}_{rb} \right\rangle, \tag{7}$$

where $g_i = \nabla_\theta \mathcal{L}_i(\theta_i^\tau(x_i^q), y_i^q)$, $g_{ri} = \gamma_i g_i = \nabla_{\hat{z}_i^q} \mathcal{L}_i(\gamma_i \theta_i^\tau(x_i^q), y_i^q)$ and $\bar{g}_{rb} = \frac{1}{N} \sum_{i \in N} g_{ri}$ is the average gradient for $\mathcal{T}_b$. Solving equation (7) is a constrained optimization problem. While this usually means using expensive algorithms like Riemannian gradient descent (Absil et al., 2009), we can leverage recent work on manifold parametrization (Casado & Martínez-Rubio, 2019) and, instead, apply unconstrained optimization methods, where R_γ is automatically parameterized via exponential maps on the Lie algebra of $SO(m)$. The optimization of the neural network in equation (3) and the RotoGBML in equations (5) and (7) can be interpreted as a Stackelberg game: a two player-game in which the leader and follower alternately move in order to minimize their respective losses, and the leader knows what will be the follower's response to their movements. Such an interpretation allows us to derive simple guidelines to guarantee training convergence, *i.e.*, the network loss does not oscillate as a result of optimizing the two different objectives. More details in Appendix C and D.

3.3 Invariant Self-Information

As mentioned above, we homogenize the task gradients from feature level. However, these features learned by neural network are susceptible to some biases, *e.g.*, image backgrounds or textures (Shi et al., 2020; LeCun et al., 2015; Brendel & Bethge, 2019). For instance, given an image with a cat's

shape filled with an elephant's skin texture, CNN tends to classify it as an elephant instead of a cat. Geirhos et al. (Geirhos et al., 2018) find that shape-biased neural network trained on stylized images are more robust to random image distortions. In this section, we further introduce the invariant self-information (ISI) module to extract the robust shape features as invariant causal features. We homogenize OOD task gradients based on these invariant causal features, which helps the reweighted vector set and the rotation matrix set avoid being affected by these biases (*e.g.*, texture features).

How to extract the robust shape features? Since there are no existing corresponding shape labels, this is a challenging problem. Given an image, we observe that the shape regions tend to be more pronounced and carry a high information compared with neighboring regions. Therefore, at each layer of the neural network, we learn shape features by saving neurons that extract high-informative regions and zeroing out neurons in low-informative regions. Specifically, we first extract features $z_i^l \in \mathbb{R}^{c^l \times k \times k}$ from input data $x_i \in \mathbb{R}^{c \times h \times w}$ in i-th task $\mathcal{T}_i$, where l is the l-th layer of the neural network. Then a drop coefficient d is proposed to drop these less-information regions as follows:

$$d(z_{i,c}^l) \propto e^{-\mathcal{I}(p_{i,k}^{l-1})/T}, \quad \text{s.t. } \mathcal{I}(p_{i,k}^{l-1}) = -\log\, q_{i,k}^{l-1}(p_{i,k}^{l-1}), \tag{8}$$

where $p_{i,k}^{l-1}$ is the k-th region for c-th channel's in z_i^{l-1} and $p_{i,k}^{l-1} \in \mathbb{R}^{c_{l-1} \times k \times k}$. $p_{i,k}^{l-1}$ is sampled from the defined distribution $q_{i,k}^{l-1}$. $\mathcal{I}$ denotes self-information and T is temperature. When the value of $\mathcal{I}$ is low, the corresponding neuron is likely to be dropped, and the network tends to rely less on $p_{i,k}^{l-1}$.

To approximate $q_{i,k}^{l-1}$, we assume that $p_{i,k}^{l-1}$ and other regions in its neighbourhood $\mathcal{N}_{i,k}^{l-1}$ come from the same distribution. Here the neighbourhood means a local region centered at $p_{i,k}^{l-1}$, with Manhattan radius C, *i.e.*, the neighbourhood contains $(2C+1)^2$ regions. Then, with neighbouring regions as samples, we approximate $q_{i,k}^{l-1}(\cdot)$ with its kernel density estimator $\hat{q}_{i,k}^{l-1}$, the representation is:

$$\hat{q}_{i,k}^{l-1}(p) = \frac{1}{(2C+1)^2} \sum\nolimits_{p' \in \mathcal{N}_{i,k}^{l-1}} K(p, p'), \tag{9}$$

where $K(\cdot,\cdot)$ is kernel function. A Gaussian kernel is used, *i.e.*, $K(p, p') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{1}{2h^2}\|p - p'\|^2)$, where h is the bandwidth. Then the information of $p_{i,k}^{l-1}$ is estimated by

$$\hat{\mathcal{I}}(p_{i,k}^{l-1}) = -\log \{\sum\nolimits_{p' \in \mathcal{N}_{i,k}^{l-1}} e^{-\frac{1}{2h^2}\|p_{i,k}^{l-1} - p'\|^2}\}. \tag{10}$$

It can be observed that the more different $p_{i,k}^{l-1}$ is from neighboring patches, the more information it contains. In other words, shapes are more unique in their surroundings and thus more informative.

3.4 Implementation

We summarize the overall proposed approach in Algorithm 1 of Appendix C. In this paper, we consider a more challenging and real-world setting, *i.e.*, tasks are randomly sampled from different distributions during training and testing. **In the training phase**, we first use the invariant self-information (ISI) module to extract the invariant causal features. Then, based on these invariant features, we use the reweighted vector set $\mathbf{w}_\omega$ to reweight task-gradient magnitudes and the rotating matrix R_γ to rotate task-gradient directions. Finally, these homogenized task gradients are used to optimize the meta-knowledge in equation 3. **In the testing phase**, invariant self-information (ISI) module is removed to examine whether our network has truly learned invariant features. We also remove $\mathbf{w}_\omega$ and R_γ to evaluate the performance of meta-knowledge learned by GBML algorithms.

4 Theoretical Analysis

In this section, we first theoretically investigate why OOD acerbates inconsistencies in task-gradient magnitudes and directions. We sample two tasks from different distributions, where $\mathcal{T}_i = (\mathcal{S}_i, \mathcal{Q}_i) \sim \mathbb{P}_i(\mathcal{T})$ and $\mathcal{T}_j = (\mathcal{S}_j, \mathcal{Q}_j) \sim \mathbb{P}_j(\mathcal{T})$ with $i \neq j$. In outer loop, the gradient difference is calculated as:

$$d_{ij} = \|\nabla_{(\phi,\theta)} \mathcal{L}_i(\phi_i^\tau(\theta_i^\tau(x_i^q)), y_i^q) - \nabla_{(\phi,\theta)} \mathcal{L}_j(\phi_j^\tau(\theta_j^\tau(x_j^q)), y_j^q)\|, \tag{11}$$

where $g_i = \nabla_{(\phi,\theta)} \mathcal{L}_i(\phi_i^\tau(\theta_i^\tau(x_i^q)), y_i^q)$ and $g_j = \nabla_{(\phi,\theta)} \mathcal{L}_j(\phi_j^\tau(\theta_j^\tau(x_j^q)), y_j^q)$ are gradients of task $\mathcal{T}_i$ and $\mathcal{T}_j$, respectively. To bridge the connections of task distributions and task gradients, we introduce *Total Variation Distance (TVD)* to re-estimate d_{ij} on the task distribution. TVD is defined as follows:

Definition 4.1. *(Total Variation Distance (TVD)) For two distributions P and Q, defined over the sample space Ω and $\sigma-$field $\mathcal{F}$, the TVD is defined as* $\|P-Q\|_{TV} := \sup_{A\in\mathcal{F}} |P(A)-Q(A)|$.

It is well-known that the total variation distance (TVD) admits the following characterization

$$\|P-Q\|_{TV} = \sup_{f:0\le f\le 1} \mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)]. \tag{12}$$

Theorem 4.1. *The difference of task gradients $d_{ij} = \|g_i - g_j\|$ can be bounded by the TVD as:*

$$d_{ij} \le 4\eta_{base} GL \left\|\mathbb{P}_i(\mathcal{T}) - \mathbb{P}_j(\mathcal{T})\right\|_{TV}, \tag{13}$$

where η_{base} is the learning rate of the inner loop. The G, L and more proofs are introduced in Appendix D. When $\mathbb{P}_i(\mathcal{T}) \neq \mathbb{P}_j(\mathcal{T})$, there is a gradient difference between $\mathcal{T}_i$ and $\mathcal{T}_j$. Then, we analyze that RotoGBML can narrow the difference between task gradients from *Cosine Theorem* (Pickover, 2009). When viewing gradients as a vector, the value of d_{ij} is simplified to calculate the length of the third edge $d_{ij} = \|g_i - g_j\|$ using *Cosine Theorem* in Figure 2 (left). Based on *Cosine Theorem*, equation (11) can be re-expressed as $\sqrt{\|g_i\|^2 + \|g_j\|^2 - 2\|g_i\|\|g_j\|\cos\langle g_i, g_j\rangle}$, where $\|\cdot\|$ means the length of the vector and $\cos\langle\cdot,\cdot\rangle$ is the consine angle between two vectors. RotoGBML homogenize task gradients by reweighting task-gradient magnitudes from $\|g_i\|$, $\|g_j\|$ to $\|g_i^*\|$, $\|g_j^*\|$, respectively and rotating task-gradient directions from $\cos\langle g_i, g_j\rangle$ to $\cos\langle g_i^*, g_j^*\rangle$ to narrow the task gradients difference from d_{ij} to d_{ij}^* in Figure 2 (right).

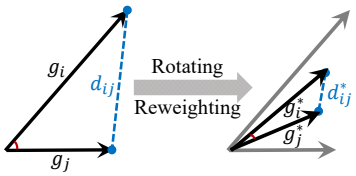


Figure 2: The difference of two OOD task gradients. (left) The task gradients for GBML. (right) we short the difference of d_{ij}^* by homogenizing magnitudes and directions.

5 Related Work

GBML frameworks for few-shot image classification. A plethora of GBML algorithms have been proposed to solve the few-shot image classification problem. The goal of these algorithms is to fast adapt to new tasks with only a few samples by transferring the meta-knowledge acquired from training tasks (Finn et al., 2017; Yoon et al., 2018; Zintgraf et al., 2019; Flennerhag et al., 2022; Lee & Choi, 2018). However, these works have a strict assumption that training and testing (or new) tasks come from the same distribution, which seriously violates the real world (see Appendix A).

Recently, some works (Lee et al., 2020; Jeong & Kim, 2020; Chen et al., 2019; Amrith et al., 2021) have been proposed to study the OOD problem of GBML. However, they mainly focus on the shifts caused by training and testing data (*e.g.*, training from *mini*ImageNet, testing from CUB). This setting follows the definition of distribution shifts in traditional machine learning (Shen et al., 2021). They do not consider the problem from the GBML optimization itself and proposed appropriate algorithms for GBML frameworks. We consider the distribution shifts at the task level due to GBML uses task information of each minibatch to optimize the meta-knowledge.

Task gradients homogenization in other fields. Homogenization of task-gradient magnitudes and directions is also found in multi-task learning (Javaloy & Valera, 2022; Chen et al., 2018; Sener & Koltun, 2018; Liu et al., 2021; Yu et al., 2020; Chen et al., 2020). Their goal is to simultaneously learn K different tasks, that is, finding K mappings from a common input dataset to a task-specific set of labels (generally K=2). Motivated by Javaloy & Valera (2022); Wang et al. (2021), we regard GBML meta-knowledge learning as a multi-task optimization process and homogenize task gradients from a multi-task perspective. Our work differ from these works in: (1) the input data is different for different tasks in GBML, but the input data of different tasks in multi-task learning is the same. (2) In multi-task learning, existing works on alleviating gradient conflicts usually assume that tasks come from the same distribution. However, we address the problem of conflicting gradients in different distributions. (3) We propose a new invariant self-information module to extract causal features.

6 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of RotoGBML and compare it with state-of-the-art algorithms. Specifically, we consider two OOD generalization problems in few-shot image classification: *The weak OOD generalization problem* is training and

Table 1: Few-shot image classification average accuracy on the weak OOD generalization problem.

Model	*mini*ImageNet		CUB	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Linear Chen et al. (2019)	42.11±0.71%	62.53±0.69%	47.12±0.74%	64.16±0.71%
ProtoNet Snell et al. (2017)	44.42±0.84%	64.24±0.72%	50.46±0.88%	76.39±0.64%
MetaOpt Lee et al. (2019b)	44.23±0.59%	63.51±0.48%	50.63±0.67%	77.11±0.59%
MTL Liu et al. (2018)	40.97±0.54%	57.12±0.68%	45.36±0.75%	74.21±0.79%
SKD-GEN1 Rajasegaran et al. (2020)	48.14±0.45%	66.36±0.36%	56.68±0.67%	76.92±0.91%
ANIL Raghu et al. (2020)	45.97±0.32%	62.10±0.45%	55.31±0.63%	75.73±0.64%
Reptile Nichol et al. (2018)	45.12±0.31%	58.92±0.41%	56.21±0.34%	72.56±0.43%
MAML Finn et al. (2017)	46.47±0.82%	62.71±0.71%	54.73±0.97%	75.75±0.76%
iMAML Rajeswaran et al. (2019)	49.30±1.88%	63.24±0.84%	56.14±0.31%	76.05±0.53%
MAML++ Antoniou et al. (2019)	47.42±0.32%	62.79±0.52%	55.19±0.34%	76.21±0.54%
ANIL-RotoGBML	46.42±0.33%	63.73±0.42%	56.42±0.32%	77.56±0.45%
Reptile-RotoGBML	47.34±0.31%	59.12±0.51%	57.87±0.32%	73.12±0.53%
MAML-RotoGBML	48.21±0.31%	65.19±0.42%	56.19±0.32%	76.21±0.45%
iMAML-RotoGBML	52.08±0.31%	66.89±0.47%	57.58±0.38%	80.98±0.44%
MAML++-RotoGBML	46.40±0.41%	63.08±0.51%	56.80±0.43%	78.72±0.52%

testing tasks from the same dataset but with disjoint classes (Triantafillou et al., 2021). *The strong OOD generalization problem* is training and testing tasks from different datasets and has multiple datasets in training data (Amrith et al., 2021; Wang & Deng, 2021; Tseng et al., 2020). Unlike these works, we use a more difficult setting, ***i.e.*, each minibatch of tasks from different datasets**.

We aim to answer the following questions: **Q1:** How does the proposed RotoGBML framework perform for the few-shot image classification task on the weak OOD generalization problem (see Section 6.1)? **Q2:** Can RotoGBML fast generalize to new tasks from new distributions when faced with the strong OOD generalization problem (see Section 6.2)? **Q3:** How well each module ($\mathbf{w}_\omega$, R_γ and ISI) in our proposed framework performs in learning the meta-knowledge (see Section 6.3)?

Datasets. For the weak OOD generalization problem, we adopt two popular benchmarks for image classification, *i.e.*, *mini*ImageNet and CUB-200-2011 (*abbr*: CUB). For the strong OOD generalization problem, we adopt nine benchmarks, including *mini*ImageNet, CUB, Cars, Places, Plantae, CropDiseases, EuroSAT, ISIC and ChestX following prior works (Tseng et al., 2020; Sun et al., 2020; Wang & Deng, 2021). In the strong OOD generalization, *mini*ImageNet is used only as training data because of its diversity, and evaluate the trained model on the other eight datasets using a leave-out-of method, *i.e.*, randomly sampled one dataset as testing data and other datasets as training data.

Backbone of GBML. Since our RotoGBML is model-agnostic and can be equipped with arbitrary GBML algorithms, we use five representative and generic GBML algorithms as our GBML backbone including MAML (Finn et al., 2017), MAML++ (Antoniou et al., 2019), ANIL (Raghu et al., 2020), Reptile (Nichol et al., 2018), iMAML (Rajeswaran et al., 2019). More details and implementations of these GBML algorithms are given in Appendix B. In all experimental results, iMAML is used as our GBML backbone due to its good performance, unless otherwise stated.

Feature encoders. We follow previous works (Chen et al., 2019; Wang & Deng, 2021) using three general networks as our feature encoder including, *i.e.* conv4 (filter:64), resnet10 and resnet18. However, it is well known that GBML algorithms (*e.g*, MAML) under-perform when applied to large networks (Mishra et al., 2018), so we use conv4 to show the main experimental results for the weak OOD generalization. For a fair comparison with some cross-domain methods (Guo et al., 2020; Sun et al., 2020; Wang & Deng, 2021; Tseng et al., 2020), we follow their setting using resnet10 for the strong OOD generalization. We also show the performance of the three feature encoders in Figure 3.

Experimental setting. We use the Adam optimizer with the learning rate of inner loop $\eta_{base} = 0.01$, outer loop $\eta_{meta} = 0.001$ and rotation matrix $\eta_\gamma = 5e-4$. We set $\beta = 0.1$ for weak and $\beta = 1.5$ for strong OOD. We optimize all models from scratch and process datasets using data augmentation following previous work (Chen et al., 2019). We evaluate the performance under two generic settings, *i.e.*, 5-way 1-shot and 5-shot and report the average accuracy as well as 95% confidence interval.

6.1 Weak OOD Generalization Problem

We conduct experiments on two representative few-shot image classification datasets: *mini*ImageNet and CUB. Table 1 reports the experimental results. From Table 1, we have some findings as follows: (1) Compared to Vanilla GBML algorithms (2-nd block), the proposed RotoGBML framework

Table 2: Few-shot image classification average accuracy on the strong OOD generalization problem.

Model	CUB		Cars		Places		Plantae		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Random Guo et al. (2020)	40.53%	53.76%	28.12%	39.21%	47.57%	61.68%	30.77%	40.45%	36.75%	48.78%
RelationNet Sung et al. (2018)	41.27%	56.77%	30.09%	40.46%	48.16%	64.25%	31.23%	42.71%	37.69%	51.05%
RelationNet-FT Tseng et al. (2020)	43.33%	59.77%	30.45%	40.18%	49.92%	65.55%	32.57%	44.29%	39.07%	52.45%
RelationNet-LRP Sun et al. (2020)	41.57%	57.70%	30.48%	41.21%	48.47%	65.35%	32.11%	43.70%	38.16%	51.99%
RelationNet-ATA Wang & Deng (2021)	43.02%	59.36%	31.79%	42.95%	51.16%	66.90%	33.72%	45.32%	39.92%	53.63%
MAML Finn et al. (2017)	40.66%	54.29%	30.02%	40.35%	45.93%	60.00%	31.35%	44.65%	36.99%	49.82%
iMAML Rajeswaran et al. (2019)	43.56%	57.98%	31.25%	41.65%	46.14%	61.52%	32.01%	43.65%	38.24%	51.20%
MAML++ Antoniou et al. (2019)	42.11%	58.72%	32.93%	43.55%	46.52%	62.61%	33.91%	42.77%	38.87%	51.91%
ANIL Raghu et al. (2020)	42.67%	55.58%	30.63%	41.77%	45.55%	61.72%	31.90%	45.95%	37.69%	51.26%
MAML-RotoGBML	41.38%	56.82%	39.19%	42.35%	46.12%	62.94%	34.21%	46.21%	40.23%	52.08%
iMAML-RotoGBML	46.12%	60.23%	35.81%	43.80%	52.65%	67.47%	35.24%	46.51%	42.46%	54.50%
MAML++-RotoGBML	44.38%	59.66%	33.66%	42.23%	50.14%	68.77%	34.14%	45.23%	40.58%	53.97%
ANIL-RotoGBML	45.86%	58.81%	31.68%	42.01%	47.20%	62.44%	32.38%	46.91%	39.28%	52.54%

Model	ChestX		CropDiseases		EuroSAT		ISIC		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Random Guo et al. (2020)	19.81%	21.80%	50.43%	69.68%	40.97%	58.00%	28.56%	37.91%	34.94%	46.85%
RelationNet Sung et al. (2018)	21.95%	24.07%	53.58%	72.86%	49.08%	65.56%	30.53%	38.60%	38.79%	50.27%
RelationNet-FT Tseng et al. (2020)	21.79%	23.95%	57.57%	75.78%	53.53%	69.13%	30.38%	38.68%	40.82%	51.89%
RelationNet-LRP Sun et al. (2020)	22.11%	24.28%	55.01%	74.21%	50.99%	67.54%	31.16%	39.97%	39.82%	51.50%
RelationNet-ATA Wang & Deng (2021)	22.14%	24.43%	61.17%	78.20%	55.69%	71.02%	31.13%	40.38%	42.53%	53.50%
MAML Finn et al. (2017)	21.72%	23.48%	52.54%	78.05%	48.10%	71.70%	28.58%	40.13%	37.74%	53.34%
iMAML Rajeswaran et al. (2019)	22.45%	24.14%	53.23%	79.23%	50.34%	71.89%	29.58%	45.65%	38.90%	55.23%
MAML++ Antoniou et al. (2019)	21.43%	22.67%	52.70%	76.45%	46.25%	72.83%	30.84%	44.51%	37.81%	54.12%
ANIL Raghu et al. (2020)	21.36%	21.83%	51.62%	77.30%	49.81%	70.67%	30.04%	45.30%	38.21%	53.78%
MAML-RotoGBML	22.92%	24.51%	57.72%	80.17%	51.93%	72.48%	30.94%	43.89%	40.88%	55.26%
iMAML-RotoGBML	24.12%	25.98%	58.34%	80.23%	55.78%	74.42%	31.25%	49.68%	42.37%	57.58%
MAML++-RotoGBML	22.97%	23.26%	55.25%	78.51%	48.44%	73.57%	32.48%	47.53%	39.79%	55.72%
ANIL-RotoGBML	23.55%	26.47%	54.88%	79.50%	50.21%	71.22%	31.21%	46.90%	39.96%	56.02%

consistently and significantly improves all performances by homogenizing task-gradient magnitudes and directions. Moreover, compared to the 95% confidence interval, it is clear that RotoGBML reduces the uncertainty of model predictions and further reduces the high variances of model learning, thereby improving the robustness and generalization. (2) Compared to some state-of-the-art few-shot image classification methods (1-st block), RotoGBML outperforms all methods, including metric-based model (ProtoNet), fine-tuning model (Linear), pretrain-based model (MTL) and knowledge-distillation-based model (SKD-GEN1). This further demonstrates the effectiveness of our RotoGBML, while also providing a new solution for GBML to outperform metric-based meta-learning methods.

We also compare the performance of the model with different feature encoders (conv4, resnet10 and resnet18) in Figure 3. We can find that (1) our RotoGBML outperforms all GBML algorithms in all experimental settings with different network sizes. (2) It's worth noting that RotoGBML can alleviate the performance degradation problem of large networks. This is mainly because our ISI module can extract invariant causal features (*i.e.*, the shapes or outlines of objects), which avoid some non-causal features to affect the generalization of large networks on some new tasks from new distributions.

6.2 Strong OOD Generalization Problem

In this section, we consider a more challenging OOD setting, *i.e.*, each task of each minibatch (typically the number of tasks $N = 4$) comes from different datasets. We compare our method and some state-of-the-art methods with the strong OOD generalization problem in Table 2. From Table 2, we have some significant findings as follows: (1) Compared to Vanilla GBML algorithms (2-nd block), RotoGBML outperforms all algorithms on eight datasets with different performance gains. These results again prove that our RotoGBML can learn robust meta-knowledge, which in turn generalizes quickly to new tasks from new distributions. (2) Compared to some cross-domain methods (1-st block), RotoGBML has a good performance in most settings, which demonstrates that we can help GBML algorithms to achieve state of the art by homogenizing task-gradient magnitudes and directions. (3) The improvement of the strong OOD setting is usually larger than that of the weak OOD setting (Table 1). The phenomenon justifies our conclusion that the larger the distribution difference, the more obvious inconsistencies in magnitudes and directions of task gradients.

In Figure 3, the performances of the strong OOD problem are shown with different network sizes. We show the results on Cars and EuroSAT as we have similar observations on other datasets. Compared to Vanilla GBML algorithms, iMAML has the best performance on the weak OOD problem, but significantly degrades on the strong OOD problem. Because it uses a regularization to encourage task

Table 3: Ablation study on all modules of RotoGBML framework. The first row represents the performance of iMAML. Violet is the weak OOD and blue is the strong OOD experiments. *hyper* means that a fixed reweighted vector set is used to reweight each task gradients in each minibatch.

R_γ	$\mathbf{w}_\omega$	ISI	*mini*ImageNet		CUB		Cars		EuroSAT	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
✗	✗	✗	49.30%	63.24%	56.14%	76.05%	31.25%	41.65%	50.34%	71.89%
✗	*hyper*	✗	48.25%(-1.1)	59.45%(-3.8)	56.78%(+0.6)	77.02%(+1.0)	30.14%(-1.1)	38.45%(-3.2)	48.46%(-1.9)	70.58%(-1.3)
✗	✓	✗	50.45%(+1.2)	64.77%(+1.5)	56.98%(+0.8)	78.39%(+2.3)	33.36%(+2.1)	43.08%(+1.4)	53.81%(+3.5)	72.65%(+0.8)
✓	✗	✗	50.42%(+1.1)	64.53%(+1.3)	57.16%(+1.0)	78.61%(+2.6)	33.59%(+2.3)	43.36%(+1.7)	53.62%(+3.3)	73.12%(+1.2)
✗	✗	✓	51.56%(+2.3)	65.17%(+2.3)	57.05%(+0.9)	77.69%(+1.6)	31.69%(+0.9)	40.04%(+0.6)	51.18%(+0.8)	72.43%(+0.5)
✓	✓	✗	50.84%(+1.5)	65.37%(+2.1)	57.47%(+1.3)	79.11%(+3.1)	34.41%(+3.2)	43.59%(+1.9)	54.21%(+3.9)	73.80%(+1.9)
✗	✓	✓	51.81%(+2.5)	66.61%(+3.4)	57.38%(+1.2)	80.56%(+4.5)	33.88%(+2.6)	43.18%(+1.5)	54.49%(+4.2)	73.43%(+1.5)
✓	✗	✓	51.95%(+2.7)	66.74%(+3.5)	57.23%(+1.1)	80.60%(+4.6)	35.01%(+3.8)	43.47%(+1.8)	54.85%(+4.5)	73.95%(+2.1)
✓	✓	✓	52.08%(+2.8)	66.89%(+3.7)	57.58%(+1.4)	80.98%(+4.9)	35.81%(+4.6)	43.80%(+2.1)	55.78%(+5.4)	74.42%(+2.5)

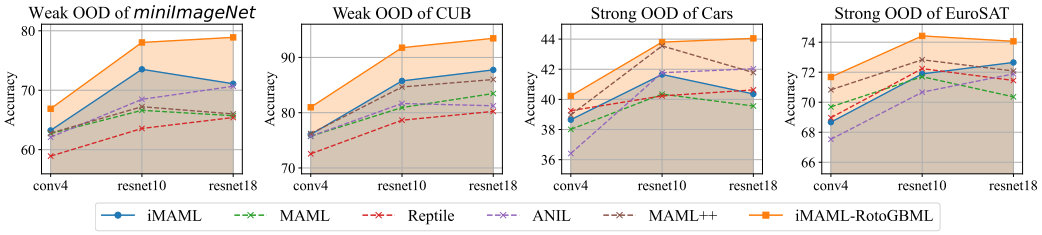


Figure 3: Few-shot image classification accuracy with different encoders for the weak and strong OOD generalization problems. iMAML-RotoGBML has the best performance in all experiments.

parameters to close to meta parameters, large distribution shifts affect the learning of meta parameters. Our RotoGBML can learn good meta parameters by homogenizing task gradients.

6.3 Ablation Study

We use a hierarchical ablation study to evaluate the performance of each module for RotoGBML under the weak and strong OOD problems in Table 3. The 1-st row represents the results of Vanilla iMAML. The plus sign means performance increase and the minus sign means performance decrease, where violet is the weak OOD and blue is the strong OOD experiments. To evaluate the performances of a fixed reweighted vector set, *e.g.*, [0.1,0.5,0.3,0.4] is used for the four tasks in each minibatch in our experiments, the results are shown in the 2-nd row. From Table 3, the following findings: (1) compared to Vanilla iMAML, the performances of hyperparameters have a significant drop in most settings, but our dynamic adjustment strategy (3-rd row) has a clear improvement. This further evaluates the effectiveness of our learning method to dynamically optimize the reweighted vectors. (2) Each module has different performance gains (3∼5-th rows). Compared to the strong OOD problem, the performances of ISI on the weak OOD problem have a large improvement, because even if invariant features are learned, the strong inconsistencies of task gradients affect the learning of meta-knowledge. (3) Arbitrary combination of two modules has good performance in 6∼8-th rows, especially with ISI. This is because adjusting the gradients based on invariant robust information from different distributions can learn better. We also give some visualization experiments of ISI learning invariant features in Appendix E. And all modules are used to have the best performance.

7 Conclusion

In this paper, we address the challenging problem of learning a good meta-knowledge of gradient-based meta-learning (GBML) algorithms under the OOD setting. We show that OOD generalization exacerbates the inconsistencies in task-gradient magnitudes and directions. Therefore, we propose RotoGBML, a general framework to dynamically reweight diverse magnitudes to a common scale and rotate conflicting directions close to each other. Moreover, we also design an invariant self-information (ISI) module to extract the invariant causal features. Homogenizing gradients using these causal features provide further guarantees for learning robust meta-knowledge. Finally, we analyze the feasibility of our method from the theoretical and experimental levels. We believe that our work makes a meaningful step toward adjusting task gradients under the OOD problem for GBML.

REFERENCES

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds. In *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

Setlur Amrith, Li Oscar, and Smith Virginia. Two sides of meta-learning evaluation: In vs. out of distribution. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2021.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *International Conference on Learning Representations, ICLR*, 2019.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

Mario Lezcano Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3794–3803. PMLR, 2019.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations, ICLR*, 2019.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, 2017.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. *International Conference on Learning Representations, ICLR*, 2022.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision, ECCV*, pp. 124–141. Springer, 2020.

Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multi-task learning. *International Conference on Learning Representations, ICLR*, 2022.

Taewon Jeong and Heeyoung Kim. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR*, pp. 7482–7491, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, NeurIPS*, 2012.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. *International Conference on Learning Representations, ICLR*, 2020.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019a.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 10657–10665, 2019b.

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

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations, ICLR*, 2021.

Q Sun Y Liu, TS Chua, and B Schiele. Meta-transfer learning for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *International Conference on Learning Representations, ICLR*, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Clifford A Pickover. *The math book: from Pythagoras to the 57th dimension, 250 milestones in the history of mathematics*. Sterling Publishing Company, Inc., 2009.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *International Conference on Learning Representations, ICLR*, 2020.

Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018.

Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. Informative dropout for robust representation learning: A shape-bias perspective. In *International Conference on Machine Learning, ICML*, pp. 8828–8839. PMLR, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems, NeurIPS*, 30, 2017.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 16519–16529, 2021.

Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek, Yunqing Zhao, Ngai-Man Cheung, and Alexander Binder. Explanation-guided training for cross-domain few-shot classification. In *International Conference on Pattern Recognition, ICPR*, 2020.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR*, pp. 1199–1208, 2018.

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

Eleni Triantafillou, Hugo Larochelle, Richard S. Zemel, and Vincent Dumoulin. Learning a universal template for few-shot dataset generalization. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations, ICLR*, 2020.

Joaquin Vanschoren. Meta-learning. In *Automated Machine Learning*, pp. 35–61. Springer, Cham, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems, NeurIPS*, 30, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2016.

Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*, 2021.

Haoxiang Wang, Han Zhao, and Bo Li. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In *Proceedings of the 38th International Conference on Machine Learning ICML*, volume 139, pp. 10991–11002. PMLR, 2021.

Jane X Wang. Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, 38:90–95, 2021.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence, TPAMI*, 2020.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. In *International Conference on Learning Representations, ICLR*, 2020.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Caml: Fast context adaptation via meta-learning. *Proceedings of the 34th International Conference on Machine Learning, ICML*, 2019.