A Simple Yet Effective Strategy to Robustify the Meta Learning Paradigm

Anonymous Author(s)
Affiliation
Address
email

Abstract

Meta learning is a promising paradigm to enable skill transfer across tasks. Most previous methods employ the empirical risk minimization principle in optimization. However, the resulting worst fast adaptation to a subset of tasks can be catastrophic in risk-sensitive scenarios. To robustify fast adaptation, this paper optimizes meta learning pipelines from a distributionally robust perspective and meta trains models with the measure of expected tail risk. We take the two-stage strategy as heuristics to solve the robust meta learning problem, controlling the worst fast adaptation cases at a certain probabilistic level. Experimental results show that our simple method can improve the robustness of meta learning to task distributions and reduce the conditional expectation of the worst fast adaptation risk.

1 Introduction

The past decade has witnessed the remarkable progress of deep learning in real-world applications (LeCun et al., 2015). However, training deep learning models requires an enormous dataset and intensive computational power. At the same time, these pre-trained models can frequently encounter deployment difficulties when the dataset's distribution drifts in testing time (Lesort et al., 2021).

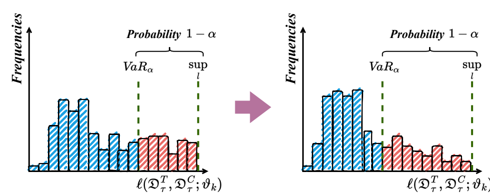

Figure 1: **Illustrations of Distributionally Robust Fast Adaptation.** Shown are histograms of meta risk function values $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ in the task distribution $p(\tau)$. Given a probability α, we optimize meta learning model parameters ϑ to decrease the risk quantity CVaR_α in Definition (3).

As a result, the paradigm of *meta learning* or *learning to learn* is proposed and impacts the machine learning scheme (Finn et al., 2017), which leverages past experiences to enable fast adaptation to unseen tasks. Moreover, in the past few years, there has grown a large body of meta learning methods to find plausible strategies to distill common knowledge into separate tasks (Finn et al., 2017; Duan et al., 2016; Garnelo et al., 2018a).

Notably, most previous work concentrates merely on the fast adaptation strategies and employs the standard risk minimization principle, *e.g.* the empirical risk minimization, ignoring the difference between tasks in fast adaptation. Given the sampled batch from the task distribution, the standard meta learning methods weight tasks equally in fast adaptation. Such an implementation raises concerns in some real-world scenarios, when worst fast adaptation is catastrophic in a range of risk-sensitive applications (Johannsmeier et al., 2019; Jaafra et al., 2019). For example, in robotic manipulations, humanoid robots (Duan, 2017) can quickly leverage past motor primitives to walk on plaint roads but might suffer from tribulation doing this on rough roads.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Research Motivations. Instead of seeking novel fast adaptation strategies, we take more interest in optimization principles for meta learning. Given the meta trained model, this paper stresses the performance difference in fast adaptation to various tasks as an indispensable consideration. As *the concept of robustness in fast adaptation* has not been sufficiently explored from the task distribution perspective, researching this topic has more practical significance and deserves more attention in meta learning. Naturally, we raise the question below:

Can we reconsider the meta learning paradigm through the lens of risk distribution, and are there plausible measures to enhance the fast adaptation robustness in some vulnerable scenarios?

Developed Methods. In an effort to address the above concerns and answer these questions, *we reduce robust fast adaptation in meta learning to a stochastic optimization problem within the principle of minimizing the expected tail risk, e.g., conditional value-at-risk (CVaR) (Rockafellar et al., 2000)*. To tractably solve the problem, we adopt a two-stage heuristic strategy for optimization with the help of crude Monte Carlo methods (Kroese and Rubinstein, 2012) and give some theoretical analysis. In each optimization step, the algorithm estimates the value-at-risk (VaR) (Rockafellar et al., 2000) from a meta batch of tasks and screens and optimizes a percentile of task samples vulnerable to fast adaptation. As illustrated in Fig. (1), such an operation is equivalent to iteratively reshaping the task risk distribution to increase robustness. The consequence of optimizing the risk function distributions $\vartheta_k \to \vartheta_{k+1}$ is to transport the probability mass in high-risk regions to the left side gradually. In this manner, the distribution of risk functions in the task domain can be optimized toward the anticipated direction that controls the worst-case fast adaptation at a certain probabilistic level.

Outline & Primary Contributions. We overview related meta learning and robust optimization work in Section (2). Section (3) introduces general notations and describes meta learning optimization objectives together with typical models. The distributionally robust meta learning problem is presented together with a heuristic optimization strategy in Section (4). We report experimental results and analysis in Section (5), followed by conclusions and limitations in Section (6). **Our primary contribution** is two-fold:

1. We recast the robustification of meta learning to a distributional optimization problem. The resulting framework minimizes the conditional expectation of task risks, namely the tail risk, which unifies vanilla meta-learning and worst-case meta learning frameworks.
2. To resolve the robust meta learning problem, we adopt the heuristic two-stage strategy and demonstrate its improvement guarantee. Experimental results show the effectiveness of our method, enhancing fast adaptation robustness and mitigating the worst-case performance.

2 Literature Review

Meta Learning Methods. In practice, meta learning enables fast learning (adaptation to unseen tasks) via slow learning (meta training in a collection of tasks). There exist different families of meta learning methods. The optimization-based methods, such as model agnostic meta learning (MAML) (Finn et al., 2017) and its variants (Finn et al., 2018; Rajeswaran et al., 2019; Grant et al., 2018; Vuorio et al., 2019; Abbas et al., 2022), try to find the optimal initial parameters of models and then execute gradient updates over them to achieve adaptation with a few examples. The context-based methods, *e.g.* conditional neural processes (CNPs) (Garnelo et al., 2018a), neural processes (NPs) (Garnelo et al., 2018b) and extensions (Gordon et al., 2019; Foong et al., 2020; Gondal et al., 2021; Wang and van Hoof, 2022; Wang et al., 2023), learn the representation of tasks in the function space and formulate meta learning models as exchangeable stochastic processes. The metrics-based methods (Snell et al., 2017; Allen et al., 2019; Bartunov and Vetrov, 2018) embed tasks in a metric space and can achieve competitive performance in few-shot classification tasks. Other methods like memory-augmented models (Santoro et al., 2016), recurrent models (Duan et al., 2016) and hyper networks (Zhao et al., 2020; Beck et al., 2023) are also modified for meta learning purposes.

Robust Optimization. When performing robust optimization for downstream tasks in deep learning, we can find massive work concerning the adversarial input noise (Goodfellow et al., 2018; Goel et al., 2020; Ren et al., 2021), or the perturbation on the model parameters (Goodfellow et al., 2014; Kurakin et al., 2016; Liu et al., 2018; Silva and Najafirad, 2020). In contrast, this paper studies the robustness of fast adaptation in meta learning. In terms of robust principles, the commonly considered one is the worst-case optimization (Olds, 2015; Zhang et al., 2020; Tay et al., 2022). For example, Collins et al. (2020) conducts the worst-case optimization in MAML to obtain the robust meta initialization. Considering the influence of adversarial examples, Goldblum et al. (2019) propose to adversarially meta train the model for few-shot image classification. Wang et al. (2020) adopt the worst-case optimization in MAML to increase the model robustness by injecting adversarial noise to the input. *However, distributionally robust optimization (Rahimian and Mehrotra, 2019) is rarely examined in the presence of the meta learning task distribution.*

3 Preliminaries

Notations. Consider the distribution of tasks $p(\tau)$ for meta learning and denote the task space by Ω_τ. Let τ be a task sampled from $p(\tau)$ with $\mathcal{T}$ the set of all tasks. We denote the meta dataset by $\mathfrak{D}_\tau$. For example, in few-shot regression problems, $\mathfrak{D}_\tau$ refers to a set of data points $\{(x_i, y_i)\}_{i=1}^m$ to fit.

Generally, $\mathfrak{D}_\tau$ are processed into the context set $\mathfrak{D}_\tau^C$ for fast adaptation, and the target set $\mathfrak{D}_\tau^T$ for evaluating adaptation performance. As an instance, we process the dataset $\mathfrak{D}_\tau = \mathfrak{D}_\tau^C \cup \mathfrak{D}_\tau^T$ with a fixed partition in MAML (Finn et al., 2017). $\mathfrak{D}_\tau^C$ and $\mathfrak{D}_\tau^T$ are respectively used for the inner loop and the outer loop in model optimization.

Definition 1 (Meta Risk Function) *With the task $\tau \in \mathcal{T}$ and the pre-processed dataset $\mathfrak{D}_\tau$ and the model parameter $\vartheta \in \Theta$, the meta risk function is a map $\ell : \mathfrak{D}_\tau \times \Theta \mapsto \mathbb{R}^+$.*

In meta learning, the meta risk function $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$, *e.g.* instantiations in Example (1)/(2), is to evaluate the model performance after fast adaptation. Now we turn to the commonly-used risk minimization principle, which plays a crucial role in fast adaptation. To summarize, we include the vanilla and worst-case optimization objectives as follows.

Expected Risk Minimization for Meta Learning. The objective that applies to most vanilla meta learning methods can be formulated in Eq. (1), and the optimization executes in a distribution over tasks $p(\tau)$. The Monte Carlo estimate corresponds to the *empirical risk minimization* principle.

$$\min_{\vartheta \in \Theta} \mathcal{E}(\vartheta) := \mathbb{E}_{p(\tau)}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\Big] \tag{1}$$

Here ϑ is the parameter of meta learning models, which includes parameters for common knowledge shared across all tasks and for fast adaptation. Furthermore, the task distribution heavily influences the direction of optimization in meta training.

Worst-case Risk Minimization for Meta Learning. This also considers meta learning in the task distribution $p(\tau)$, but the worst case in fast adaptation is the top priority in optimization.

$$\min_{\vartheta \in \Theta} \max_{\tau \in \mathcal{T}} \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \tag{2}$$

The optimization objective is built upon the min-max framework, advancing the robustness of meta learning to the worst case. Approaches like TR-MAML (Collins et al., 2020) sample the worst task in a batch to meta train with gradient updates. Nevertheless, this setup might result in a highly conservative solution where the worst case only happens with an incredibly lower chance.

4 Distributionally Robust Fast Adaptation

This section starts with the concept of risk measures and the derived meta learning optimization objective. Then a heuristic strategy is designed to approximately solve the problem. Finally, we provide two examples of distributionally robust meta learning methods.

4.1 Meta Risk Functions as Random Variables

Assumption 1 *The meta risk function $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ is β_τ-Lipschitz continuous w.r.t. ϑ, which suggests: there exists a positive constant β_τ such that $\forall\{\vartheta, \vartheta'\} \in \Theta$:*

$$|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')| \leq \beta_\tau ||\vartheta - \vartheta'||.$$

Let $(\Omega_\tau, \mathcal{F}_\tau, \mathbb{P}_\tau)$ denote a probability measure over the task space, where $\mathcal{F}_\tau$ corresponds to a σ-algebra on the subsets of Ω_τ. And we have $(\mathbb{R}^+, \mathcal{B})$ a probability measure over the non-negative real domain for the previously mentioned meta risk function $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ with $\mathcal{B}$ a Borel σ-algebra. For any $\vartheta \in \Theta$, the meta learning operator $\mathcal{M}_\vartheta : \Omega_\tau \mapsto \mathbb{R}^+$ is defined as:

$$\mathcal{M}_\vartheta : \tau \mapsto \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta).$$

In this way, $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ can be viewed as a random variable to induce the distribution $p(\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta))$. Further, the cumulative distribution function can be formulated as $F_\ell(l; \vartheta) = \mathbb{P}(\{\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \leq l; \tau \in \Omega_\tau, l \in \mathbb{R}^+\})$ *w.r.t.* the task space. Note that $F_\ell(l; \vartheta)$ implicitly depends on the model parameter ϑ, and we cannot access a closed-form in practice.

Definition 2 (Value-at-Risk) *Given the confidence level $\alpha \in [0, 1]$, the task distribution $p(\tau)$ and the model parameter ϑ, the VaR (Rockafellar et al., 2000) of the meta risk function is defined as:*

$$\text{VaR}_\alpha [\ell(\mathcal{T}, \vartheta)] = \inf_{l \in \mathbb{R}^+} \{l | F_\ell(l; \vartheta) \geq \alpha, \tau \in \mathcal{T}\}.$$

Definition 3 (Conditional Value-at-Risk) *Given the confidence level $\alpha \in [0, 1]$, the task distribution $p(\tau)$ and the model parameter ϑ, we focus on the constrained domain of the random variable $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ with $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \geq \text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)]$. The conditional expectation of this is termed as conditional value-at-risk (Rockafellar et al., 2000):*

$$\text{CVaR}_\alpha [\ell(\mathcal{T}, \vartheta)] = \int_0^\infty l dF_\ell^\alpha(l; \vartheta),$$

where the normalized cumulative distribution is as follows:

$$F_\ell^\alpha(l; \vartheta) = \begin{cases} 0, & l < \text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)] \\ \frac{F_\ell(l;\vartheta) - \alpha}{1 - \alpha}, & l \geq \text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)]. \end{cases}$$

This results in the normalized probability measure $(\Omega_{\alpha,\tau}, \mathcal{F}_{\alpha,\tau}, \mathbb{P}_{\alpha,\tau})$ over the task space, where $\Omega_{\alpha,\tau} := \bigcup_{\ell \geq \text{VaR}_\alpha[\ell(\mathcal{T},\vartheta)]} \left[\mathcal{M}_\vartheta^{-1}(\ell)\right]$. For ease of presentation, we denote the corresponding task distribution constrained in $\Omega_{\alpha,\tau}$ by $p_\alpha(\tau; \vartheta)$.

Rather than optimizing VaR_α, a quantile, in meta learning, we take more interest in CVaR_α optimization, a type of *the expected tail risk*. Such risk measure regards the conditional expectation and has more desirable properties for meta learning: *more adequate in handling adaptation risks in extreme tails*, *more accessible sensitivity analysis w.r.t.* α, and *more efficient optimization*.

Remark 1 *$\text{CVaR}_\alpha [\ell(\mathcal{T}, \vartheta)]$ to minimize is respectively equivalent with the vanilla meta learning optimization objective in Eq. (1) when $\alpha = 0$ and the worst-case meta learning optimization objective in Eq. (3) when α is sufficiently close to 1.*

4.2 Meta Learning via Controlling the Expected Tail Risk

As mentioned in Remark (1), the previous two meta learning objectives can be viewed as special cases within the CVaR_α principle. Furthermore, we turn to a particular distributionally robust fast adaptation with the adjustable confidence level α to control the expected tail risk in optimization as follows.

Distributionally Robust Meta Learning Objective. With the previously introduced normalized probability density function $p_\alpha(\tau; \vartheta)$, minimizing $\text{CVaR}_\alpha [\ell(\mathcal{T}, \vartheta)]$ can be rewritten as Eq. (3).

$$\min_{\vartheta \in \Theta} \mathcal{E}_\alpha(\vartheta) := \mathbb{E}_{p_\alpha(\tau;\vartheta)} \left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\right] \quad (3)$$

Even though $\text{CVaR}_\alpha[\ell(\mathcal{T},\vartheta)]$ is a function of the model parameter ϑ, the integral in Eq. (3) is intractable due to the involvement of $p_\alpha(\tau;\vartheta)$ in a non-closed form.

Assumption 2 *For meta risk function values, the cumulative distribution function $F_\ell(l;\vartheta)$ is β_ℓ-Lipschitz continuous w.r.t. l, and the implicit normalized probability density function of tasks $p_\alpha(\tau;\vartheta)$ is β_θ-Lipschitz continuous w.r.t. ϑ.*

Assumption 3 *For any valid $\vartheta \in \Theta$ and corresponding implicit normalized probability density function of tasks $p_\alpha(\tau;\vartheta)$, the meta risk function value can be bounded by a positive constant $\mathcal{L}_{\max}$:*

$$\sup_{\tau\in\Omega_{\alpha,\tau}} \ell(\mathfrak{D}_{\tau_i}^T,\mathfrak{D}_{\tau_i}^C;\vartheta) \leq \mathcal{L}_{\max}.$$

Proposition 1 *Under assumptions (1)/(2)/(3), the meta learning optimization objective $\mathcal{E}_\alpha(\vartheta)$ in Eq. (3) is continuous w.r.t. ϑ.*

Further, we use $\xi_\alpha(\vartheta)$ to denote the $\text{VaR}_\alpha[\ell(\mathcal{T},\vartheta)]$ for simple notations. The same as that in (Rockafellar et al., 2000), we introduce a slack variable $\xi \in \mathbb{R}$ and the auxiliary risk function $\left[\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)-\xi\right]^+ := \max\{\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)-\xi,0\}$. To circumvent directly optimizing the non-analytical $p_\alpha(\tau;\vartheta)$, we can convert the probability constrained function $\mathcal{E}_\alpha(\vartheta)$ to the below unconstrained one after optimizing ξ:

$$\varphi_\alpha(\xi;\vartheta) = \xi + \frac{1}{1-\alpha}\mathbb{E}_{p(\tau)}\left[\left[\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)-\xi\right]^+\right].$$

It is demonstrated that $\mathcal{E}_\alpha(\vartheta) = \min_{\xi\in\mathbb{R}}\varphi_\alpha(\xi;\vartheta)$ and $\xi_\alpha \in \arg\min_{\xi\in\mathbb{R}}\varphi_\alpha(\xi;\vartheta)$ in (Rockafellar et al., 2000), and also note that CVaR_α is the upper bound of ξ_α, implying

$$\xi_\alpha \leq \varphi_\alpha(\xi_\alpha;\vartheta) \leq \varphi_\alpha(\xi;\vartheta), \quad \forall\xi\in\mathbb{R} \text{ and } \forall\vartheta\in\Theta. \tag{4}$$

With the deductions from Eq.s (3)/(4), we can resort the distributionally robust meta learning optimization objective with the probability constraint into a unconstrained optimization objective as Eq.(5).

$$\min_{\vartheta\in\Theta,\xi\in\mathbb{R}} \xi + \frac{1}{1-\alpha}\mathbb{E}_{p(\tau)}\left[\left[\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)-\xi\right]^+\right] \tag{5}$$

Sample Average Approximation. For the stochastic programming problem above, it is mostly intractable to derive the analytical form of the integral. Hence, we need to perform Monte Carlo estimates of Eq. (5) to obtain Eq. (6) for optimization.

$$\min_{\vartheta\in\Theta,\xi\in\mathbb{R}} \xi + \frac{1}{(1-\alpha)\mathcal{B}}\sum_{i=1}^{\mathcal{B}}\left[\ell(\mathfrak{D}_{\tau_i}^T,\mathfrak{D}_{\tau_i}^C;\vartheta)-\xi\right]^+ \tag{6}$$

Remark 2 *If $\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)$ is convex w.r.t. ϑ, then Eq.s (5)/(6) are also convex functions. In this case, the optimization objective Eq. (6) of our interest can be resolved with the help of several convex programming algorithms (Fan et al., 2017; Meng et al., 2020; Levy et al., 2020).*

4.3 Heuristic Algorithms for Optimization

Unfortunately, most existing meta learning models' risk functions (Finn et al., 2017; Garnelo et al., 2018a; Santoro et al., 2016; Li et al., 2017; Duan et al., 2016), $\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)$ are non-convex *w.r.t.* ϑ, bringing difficulties in optimization of Eq.s (5)/(6).

To this end, we propose a simple yet effective optimization strategy, where the sampled task batch is used to approximate the VaR_α and the integral in Eq. (5) for deriving Eq. (6). In detail, two stages are involved in iterations: *(i) approximate $\text{VaR}_\alpha[\ell(\mathcal{T},\vartheta)] \approx \hat{\xi}_\alpha$ with the meta batch values, which can be achieved via either a quantile estimator (Dong and Nakayama, 2018) or other density estimators; (ii) optimize ϑ in Eq. (6) via stochastic updates after replacing ξ_α by the estimated $\hat{\xi}_\alpha$.*

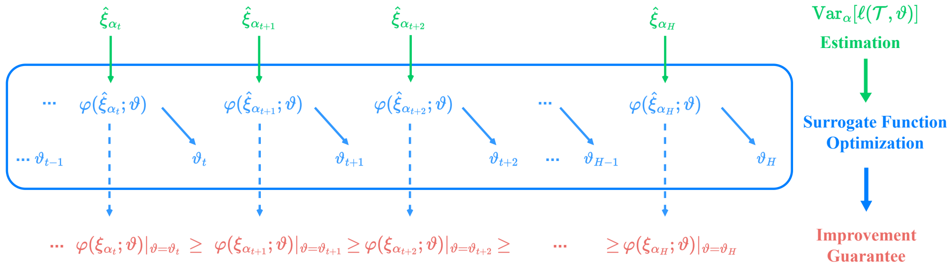

Figure 2: **Optimization Diagram of Distributionally Robust Meta Learning with Surrogate Functions.** From left to right: the meta model parameters ϑ in the middle block are optimized *w.r.t.* the constructed surrogate function $\varphi(\hat{\xi}_{\alpha_t};\vartheta)$ marked in blue in t-th iteration. Under certain conditions in Theorem (1), the distributionally robust meta learning objective $\varphi(\xi_\alpha;\vartheta)$ marked in pink can be decreased monotonically until it reaches the convergence in the H-th iteration.

Proposition 2 *Suppose there exists $\delta \in \mathbb{R}^+$ such that $|\xi_\alpha(\vartheta) - \hat{\xi}_\alpha(\vartheta)| < \delta$ with $\hat{\xi}_\alpha(\vartheta)$ an estimate of $\xi_\alpha(\vartheta)$. Then there exists a constant $\kappa_\alpha = \max\{\frac{2-\alpha}{1-\alpha}, \frac{\alpha}{1-\alpha}\}$ such that*

$$\varphi_\alpha(\hat{\xi}_\alpha(\vartheta);\vartheta) - \kappa_\alpha\delta < \mathcal{E}_\alpha(\vartheta) \leq \varphi_\alpha(\hat{\xi}_\alpha(\vartheta);\vartheta).$$

The performance gap resulting from VaR_α approximation error is estimated in Proposition (2). For ease of implementation, we adopt crude Monte Carlo methods (Kroese and Rubinstein, 2012) to obtain a consistent estimator of ξ_α.

Theorem 1 (Improvement Guarantee) *Under assumptions (1)/(2)/(3), suppose that the estimate error with the crude Monte Carlo holds: $|\hat{\xi}_{\alpha_t} - \xi_{\alpha_t}| \leq \frac{\lambda}{\beta_\ell(1-\alpha)^2}, \forall t \in \mathbb{N}^+$, with the subscript t the iteration number, λ the learning rate in stochastic gradient descent, β_ℓ the Lipschitz constant of the risk cumulative distribution function, and α the confidence level. Then the proposed heuristic algorithm with the crude Monte Carlo can produce at least a local optimum for distributionally robust fast adaptation.*

Note that Theorem (1) indicates that under certain conditions, using the above heuristic algorithm has the performance improvement guarantee, which corresponds to Fig. (2). The error resulting from the approximate algorithm is further estimated in Appendix Theorem (2).

4.4 Instantiations & Implementations

Our proposed optimization strategy applies to all meta learning methods and has the improvement guarantee in Theorem (1). Here we take two representative meta learning methods, MAML (Finn et al., 2017) and CNP (Garnelo et al., 2018a), as examples and show how to robustify them through the lens of risk distributions. Note that the forms of $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C;\vartheta)$ sometime differ in these methods. Also, the detailed implementation of the strategy relies on specific meta learning algorithms and models.

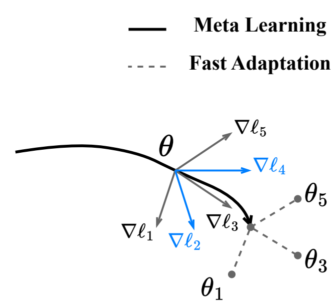

Figure 3: **Diagram of Distributionally Robust Fast Adaptation for Model Agnostic Meta Learning (Finn et al., 2017).** For example, with the size of the meta batch 5 and $\alpha = 40\%$, $5*(1-\alpha)$ tasks in gray with the worst fast adaptation performance are screened for updating meta initialization.

Example 1 (DR-MAML) *With the task distribution $p(\tau)$ and model agnostic meta learning (Finn et al., 2017), the distributionally robust MAML treats the meta learning problem as a bi-level optimization with a VaR_α relevant constraint.*

$$\min_{\substack{\vartheta\in\Theta\\ \xi\in\mathbb{R}}} \xi + \frac{1}{1-\alpha}\mathbb{E}_{p(\tau)}\left[\left[\ell(\mathfrak{D}_\tau^T;\vartheta - \lambda\nabla_\vartheta\ell(\mathfrak{D}_\tau^C;\vartheta)) - \xi\right]^+\right] \tag{7}$$

The gradient operation $\nabla_\vartheta\ell(\mathfrak{D}_\tau^C;\vartheta)$ corresponds to the inner loop with the learning rate λ.

The resulting distributionally robust MAML (DR-MAML) is still a optimization-based method, where a fixed percentage of tasks are screened for the outer loop. As shown in Eq. (7), the objective is to obtain a robust meta initialization of the model parameter ϑ.

Example 2 (DR-CNP) *With the task distribution $p(\tau)$ and the conditional neural process (Garnelo et al., 2018a), the distributionally robust conditional neural process learns the functional representations with a CVaR$_\alpha$ constraint.*

$$\begin{aligned} \min_{\substack{\vartheta\in\Theta \\ \xi\in\mathbb{R}}} \xi + \frac{1}{1-\alpha}\mathbb{E}_{p(\tau)}\left[\left[\ell(\mathfrak{D}_\tau^T; z, \theta_2)) - \xi\right]^+\right] \\ s.t.\ z = h_{\theta_1}(\mathfrak{D}_\tau^C) \text{ with } \vartheta = \{\theta_1, \theta_2\} \end{aligned} \tag{8}$$

Here h_{θ_1} is a set encoder network with θ_2 the parameter of the decoder network.

The resulting distributionally robust CNP (DR-CNP) is to find a robust functional embedding to induce underlying stochastic processes. Still, in Eq. (8), a proportion of tasks with the worst functional representation performance are used in meta training.

Moreover, we convey the pipelines of optimizing these developed distributionally robust models in Appendix **Algorithms (1)/(2)**.

5 Experimental Results and Analysis

This section presents experimental results and examines fast adaptation performance in a distributional sense. Without loss of generality, we take DR-MAML in Example (1) to run experiments.

Benchmarks. The same as work in (Collins et al., 2020), we use two commonly-seen downstream tasks for meta learning experiments: few-shot regression and image classification. Besides, ablation studies are included to assess other factors' influence or the proposed strategy's scalability.

Baselines & Evaluations. Since the primary investigation is regarding risk minimization principles, we consider the previously mentioned *expected risk minimization*, *worst-case minimization*, and *expected tail risk minimization* for meta learning. Hence, MAML (empirical risk), TR-MAML (worst-case risk), and DR-MAML (expected tail risk) serve as examined methods. We evaluate these methods' performance based on the *Average*, *Worst-case*, and *CVaR$_\alpha$* metrics. For the confidence level to meta train DR-MAML, we empirically set $\alpha = 0.7$ for few-shot regression tasks and $\alpha = 0.5$ image classification tasks without external configurations.

5.1 Sinusoid Regression

Following (Finn et al., 2017; Collins et al., 2020), we conduct experiments in sinusoid regression tasks. The mission is to approximate the function $f(x) = a\sin(x-b)$ with `K-shot` randomly sampled function points, where the task is defined by a and b. In the sine function family, the target range, amplitude range, and phase range are respectively $[-5.0, 5.0] \subset \mathbb{R}$, $a \in [0.1, 5.0]$ and $b \in [0, 2\pi]$. In the setup of meta training and testing datasets, a distributional shift exists: numerous easy tasks and several difficult tasks are generated to formulate the training dataset with all tasks in the space as the testing one. Please refer to Appendix (J) for a detailed partition of meta-training, testing tasks, and neural architectures.

Table 1: **Test average mean square errors (MSEs) with reported standard deviations for sinusoid regression (5 runs).** We respectively consider `5-shot` and `10-shot` cases with $\alpha = 0.7$. The results are evaluated across the 490 meta-test tasks, as in (Collins et al., 2020). The best results are in bold.

Method	`5-shot`			`10-shot`		
	Average	Worst	CVaR$_\alpha$	Average	Worst	CVaR$_\alpha$
MAML (Finn et al., 2017)	1.02±0.10	3.89±0.83	2.25±0.15	0.66±0.16	2.57±0.70	1.15±0.19
TR-MAML (Collins et al., 2020)	1.09±0.08	**2.28±0.35**	1.79±0.06	0.77±0.11	**1.68±0.43**	1.27±0.28
DR-MAML (Ours)	**0.89±0.04**	2.91±0.46	**1.76±0.02**	**0.54±0.01**	1.70±0.17	**0.96±0.01**

Result Analysis. We list meta-testing MSEs in sinusoid regression in Table (1). As expected, the tail risk minimization principle in DR-MAML can lead to an intermediate performance in the worst-case. In both cases, the comparison between MAML and DR-MAML in MSEs indicates that such probabilistic-constrained optimization in the task space even has the potential to advance average fast adaptation performance. In contrast, TR-MAML has to sacrifice more average performance for the worst-case improvement.

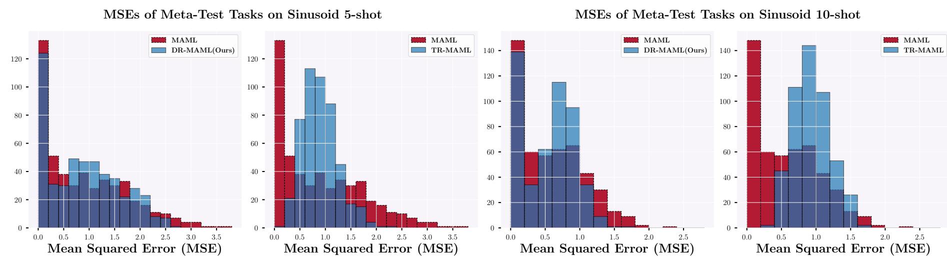

Figure 4: **Histograms of Meta-Testing Performance in Sinusoid Regression Problems.** With $\alpha = 0.7$, we respectively visualize the comparison results, DR-MAMAL-vs-MAML and TR-MAML-vs-MAML in `5-shot` (Two Sub-figures Left Side) and `10-shot` (Two Sub-figures Right Side) cases, for a sample trial.

More intuitively, Fig. (4) illustrates MSE statistics on the testing task distribution and further verifies the effect of the CVaR_α principle in decreasing the proportional worst-case errors. In `5-shot` cases, the difference in MSE statistics is more significant: *the worst-case method tends to increase the skewness in risk distributions* with many task risk values gathered in regions of intermediate performance, which is unfavored in general cases. As for why DR-MAML surpasses MAML in terms of average performance, we attribute it to the benefits of external robustness in several scenarios, *e.g.*, the drift of training/testing task distributions.

Table 2: **Average `N-way K-shot` classification accuracies in Omniglot with reported standard deviations (3 runs).** With $\alpha = 0.5$, the best results are in bold.

	Meta-Training Alphabets						Meta-Testing Alphabets					
	`5-way 1-shot`			`20-way 1-shot`			`5-way 1-shot`			`20-way 1-shot`		
Method	Average	Worst	CVaR_α	Average	Worst	CVaR_α	Average	Worst	CVaR_α	Average	Worst	CVaR_α
MAML (Finn et al., 2017)	**98.4**±0.2	82.4±1.1	**96.9**±0.5	99.2±0.1	33.9±3.0	80.9±0.7	93.5±0.2	82.5±0.2	91.6±0.6	67.6±2.0	49.7±3.5	60.4±1.7
TR-MAML (Collins et al., 2020)	97.4±0.6	**95.0**±0.3	96.5±0.4	92.2±0.8	**82.4**±2.1	**87.2**±0.9	93.1±1.1	**85.3**±1.9	91.3±0.9	74.3±1.4	**58.4**±1.8	**68.5**±1.2
DR-MAML (Ours)	97.1±0.3	84.0±0.4	95.1±0.3	**99.6**±0.6	57.9±2.4	84.8±0.7	**93.7**±0.4	84.1±0.8	**92.1**±0.5	**74.6**±1.2	51.0±2.3	66.4±1.4

5.2 Few-Shot Image Classification

Table 3: **Average `5-way 1-shot` classification accuracies in *mini*-ImageNet with reported standard deviations (3 runs).** With $\alpha = 0.5$, the best results are in bold.

	Eight Meta-Training Tasks			Four Meta-Testing Tasks		
Method	Average	Worst	CVaR_α	Average	Worst	CVaR_α
MAML (Finn et al., 2017)	70.1±2.2	48.0±4.5	63.2±2.6	46.6±0.4	44.7±0.7	44.6±0.7
TR-MAML (Collins et al., 2020)	63.2±1.3	60.7±1.6	62.1±1.2	48.5±0.6	45.9±0.8	46.6±0.5
DR-MAML (Ours)	**70.2**±0.2	**63.4**±0.2	**67.2**±0.1	**49.4**±0.1	**47.1**±0.1	**47.5**±0.1

Here we do investigations in few-shot image classification. Each task is an `N-way K-shot` classification with `N` the number of classes and `K` the number of labeled examples in one class. The Omniglot (Lake et al., 2015) and *mini*-ImageNet (Vinyals et al., 2016) datasets work as benchmarks for examination. We retain the setup of datasets in work (Collins et al., 2020).

Result Analysis. The classification accuracies in Omniglot are illustrated in Table (2): In `5-way 1-shot` cases, DR-MAML obtains the best average and CVaR_α in meta-testing datasets, while TR-MAML achieves the best worst-case performance with a slight degradation of average performance compared to MAML in both training/testing datasets. In `20-way 1-shot` cases, for training/testing datasets, we surprisingly notice that the expected tail risk is not well optimized with DR-MAML, but there is an average performance gain; while TR-MAML works best in worst-case/CVaR_α metrics.

When it comes to *mini*-ImageNet, findings are distinguished a lot from the above one: in Table (3), DR-MAML yields the best result in all evaluation metrics and cases, even regarding the worst-case. TR-MAML can also improve all metrics in meta-testing cases. Overall, challenging meta-learning tasks can reveal more advantages of DR-MAML over others.

5.3 Ablation Studies

This part mainly checks the influence of the confidence level α and the meta batch size towards the distribution of fast adapted risk values. Apart from examining these factors of interest, additional experiments are also conducted in this paper; please refer to Appendix (K) for more details.

Sensitivity to Confidence Levels α. To deepen understanding of the confidence level α's effect in fast adaptation performance, we vary α to train DR-MAML and evaluate models under previously

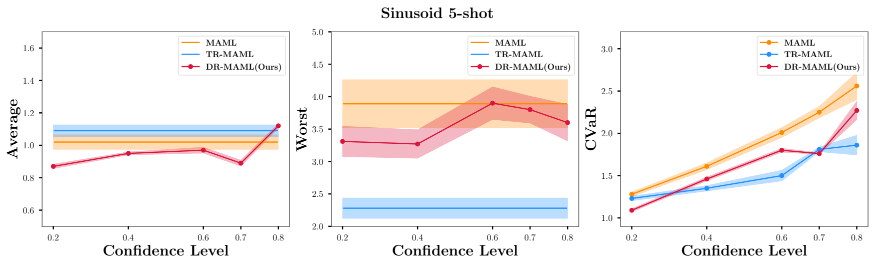

Figure 5: **Meta Testing MSEs of Meta-Trained DR-MAML with Various Confidence Levels α.** MAML and TR-MAML are irrelevant with the variation of α in meta-training. The plots report testing MSEs with standard error bars in shadow regions.

mentioned metrics. Taking the sinusoid `5-shot` regression as an example, we calculate MSEs and visualize the fluctuations with additional α-values in Fig. (5). The results in the worst-case exhibit higher deviations. The trend in Average/CVaR$_\alpha$ metrics shows that with increasing α, DR-MAML gradually approaches TR-MAML in average and CVaR$_\alpha$ MSEs. With $\alpha \leq 0.8$, ours is less sensitive to the confidence level and mostly beats MAML/TR-MAML in average performance. Though ours aims at optimizing CVaR$_\alpha$, it cannot always ensure such a metric to surpass TR-MAML in all confidence levels due to rigorous assumptions in Theorem (1).

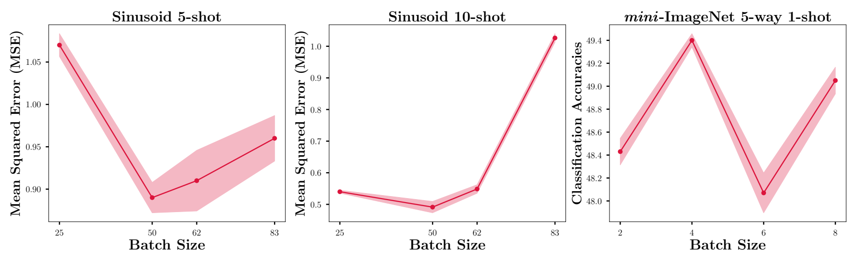

Figure 6: **Meta Testing Average Performance of Meta-Trained DR-MAML with Various Sizes of the Meta Batch.** The plots report average results with standard error bars in shadow regions.

Influence of the Task Batch Size. Note that our optimization strategy relies on the estimate of VaR$_\alpha$, and the improvement guarantee relates to the estimation error. Theoretically, the meta batch in training can directly influence the optimization result. Here we vary the meta batch size in training, and evaluated results are visualized in Fig. (6). In regression scenarios, the average MSEs can decrease to a certain level with an increase of meta batch, and then performance degrades. We attribute the performance gain with a batch increase to more accurate VaR$_\alpha$ estimates; however, a meta batch larger than some threshold can worsen the first-order meta learning algorithms' efficiency, similarly observed in (Nichol et al., 2018). As for classification scenarios, there appears no clear trend since the meta batch is smaller enough.

6 Conclusion and Limitations

Technical Discussions. This work contributes more insights into robustifying fast adaptation in meta learning. Our utilized expected tail risk trades off the expected risk minimization and worst-case risk minimization, and the two-stage strategy works as the heuristic to approximately solve the problem with an improvement guarantee. Our strategy can empirically alleviate the worst-case fast adaptation and sometimes even improve average performance.

Existing Limitations. Though our robustification strategy is simple yet effective in implementations, empirical selection of the optimal meta batch size is challenging, especially for first-order optimization methods. Meanwhile, the theoretical analysis only applies to a fraction of meta learning tasks when risk function values are in a compact continuous domain.

Future Extensions. Designing a heuristic algorithm with an improvement guarantee is non-trivial and relies on the properties of risk functions. This research direction has practical meaning in the era of large models and deserves more investigations in terms of optimization methods. Also, establishing connections between the optimal meta batch size and specific stochastic optimization algorithms can be a promising theoretical research issue in this domain.

References

Abbas, M., Xiao, Q., Chen, L., Chen, P.-Y., and Chen, T. (2022). Sharp-maml: Sharpness-aware model-agnostic meta learning. In *International Conference on Machine Learning*, pages 10–32. PMLR.

Allen, K., Shelhamer, E., Shin, H., and Tenenbaum, J. (2019). Infinite mixture prototypes for few-shot learning. In *International Conference on Machine Learning*, pages 232–241. PMLR.

Antoniou, A., Edwards, H., and Storkey, A. (2019). How to train your maml. In *Seventh International Conference on Learning Representations*.

Bahadur, R. R. (1966). A note on quantiles in large samples. *The Annals of Mathematical Statistics*, 37(3):577–580.

Barabás, B. (1987). Estimation of density functions by order statistics. *Periodica Mathematica Hungarica*, 18(2):115–122.

Bartunov, S. and Vetrov, D. (2018). Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pages 670–678. PMLR.

Beck, J., Jackson, M. T., Vuorio, R., and Whiteson, S. (2023). Hypernetworks in meta-reinforcement learning. In *Conference on Robot Learning*, pages 1478–1487. PMLR.

Chen, R. S., Lucier, B., Singer, Y., and Syrgkanis, V. (2017). Robust optimization for non-convex objectives. *Advances in Neural Information Processing Systems*, 30.

Collins, L., Mokhtari, A., and Shakkottai, S. (2020). Task-robust model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 33:18860–18871.

Dong, H. and Nakayama, M. K. (2018). A tutorial on quantile estimation via monte carlo. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 3–30. Springer.

Duan, Y. (2017). *Meta learning for control*. University of California, Berkeley.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Duchi, J. C. and Namkoong, H. (2021). Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2021). Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *Advances in Neural Information Processing Systems*, 34:5469–5480.

Fan, Y., Lyu, S., Ying, Y., and Hu, B. (2017). Learning with average top-k loss. *Advances in neural information processing systems*, 30.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Finn, C., Xu, K., and Levine, S. (2018). Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31.

Foong, A., Bruinsma, W., Gordon, J., Dubois, Y., Requeima, J., and Turner, R. (2020). Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295.

Gagne, C. and Dayan, P. (2021). Two steps to risk sensitivity. *Advances in Neural Information Processing Systems*, 34:22209–22220.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. A. (2018a). Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. (2018b). Neural processes. *arXiv preprint arXiv:1807.01622*.

Goel, A., Agarwal, A., Vatsa, M., Singh, R., and Ratha, N. K. (2020). Dndnet: Reconfiguring cnn for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 22–23.

Goldblum, M., Fowl, L., and Goldstein, T. (2019). Robust few-shot learning with adversarially queried meta-learners.

Gondal, M. W., Joshi, S., Rahaman, N., Bauer, S., Wuthrich, M., and Scholkopf, B. (2021). Function contrastive learning of transferable meta-representations. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3755–3765. PMLR.

Goodfellow, I., McDaniel, P., and Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. (2019). Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. (2018). Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*.

Hiraoka, T., Imagawa, T., Mori, T., Onishi, T., and Tsuruoka, Y. (2019). Learning robust options by conditional value at risk optimization. *Advances in Neural Information Processing Systems*, 32.

Hsieh, Y.-P., Mertikopoulos, P., and Cevher, V. (2021). The limits of min-max optimization algorithms: Convergence to spurious non-critical sets. In *International Conference on Machine Learning*, pages 4337–4348. PMLR.

Jaafra, Y., Laurent, J. L., Deruyver, A., and Naceur, M. S. (2019). Robust reinforcement learning for autonomous driving.

Jiang, Y., Li, C., Dai, W., Zou, J., and Xiong, H. (2021). Monotonic robust policy optimization with model discrepancy. In *International Conference on Machine Learning*, pages 4951–4960. PMLR.

Johannsmeier, L., Gerchow, M., and Haddadin, S. (2019). A framework for robot manipulation: Skill formalism, meta learning and adaptive control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5844–5850. IEEE.

Juditsky, A., Nemirovski, A., and Tauvel, C. (2011). Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58.

Kroese, D. P. and Rubinstein, R. Y. (2012). Monte carlo methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):48–58.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Larochelle, S. (2017). Optimization as a model for few-shot learning. In *International Conference on Learning Representations*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lesort, T., Caccia, M., and Rish, I. (2021). Understanding continual learning settings with data distribution drift analysis. *arXiv preprint arXiv:2104.01678*.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. (2020). Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860.

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

Liu, Q., Liu, T., Liu, Z., Wang, Y., Jin, Y., and Wen, W. (2018). Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 721–726. IEEE.

Meng, S. Y., Charisopoulos, V., and Gower, R. M. (2020). A stochastic prox-linear method for cvar minimization.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Stochastic approximation approach to stochastic programming. *SIAM J. Optim. Citeseer*.

Nichol, A., Achiam, J., and Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Olds, K. C. (2015). Global indices for kinematic and force transmission performance in parallel robots. *IEEE Transactions on Robotics*, 31(2):494–500.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. (2017). Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR.

Quaranta, A. G. and Zaffaroni, A. (2008). Robust optimization of conditional value at risk and portfolio selection. *Journal of Banking & Finance*, 32(10):2046–2056.

Rahimian, H. and Mehrotra, S. (2019). Distributionally robust optimization: A review.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32.

Ren, J., Zhang, D., Wang, Y., Chen, L., Zhou, Z., Chen, Y., Cheng, X., Wang, X., Zhou, M., Shi, J., et al. (2021). Towards a unified game-theoretic view of adversarial perturbations and robustness. *Advances in Neural Information Processing Systems*, 34:3797–3810.

Rockafellar, R. T., Uryasev, S., et al. (2000). Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42.

Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. *Proceedings of the national Academy of Sciences*, 42(1):43–47.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2020). Distributionally robust neural networks. In *International Conference on Learning Representations*.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

Shalev-Shwartz, S. and Wexler, Y. (2016). Minimizing the maximal loss: How and why. In *International Conference on Machine Learning*, pages 793–801. PMLR.

Silva, S. H. and Najafirad, P. (2020). Opportunities and challenges in deep learning adversarial robustness: A survey. *arXiv preprint arXiv:2007.00753*.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Tay, S. S., Foo, C. S., Daisuke, U., Leong, R., and Low, B. K. H. (2022). Efficient distributionally robust bayesian optimization with worst-case sensitivity. In *International Conference on Machine Learning*, pages 21180–21204. PMLR.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., and Manzagol, P. A. (2019). Meta-dataset: A dataset of datasets for learning to learn from few examples.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. (2019). Multimodal model-agnostic meta-learning via task-aware modulation. *Advances in neural information processing systems*, 32.

Wang, Q., Federici, M., and van Hoof, H. (2023). Bridge the inference gaps of neural processes via expectation maximization. In *The Eleventh International Conference on Learning Representations*.

Wang, Q. and van Hoof, H. (2022). Learning expressive meta-representations with mixture of expert neural processes. In *Advances in neural information processing systems*.

Wang, R., Xu, K., Liu, S., Chen, P.-Y., Weng, T.-W., Gan, C., and Wang, M. (2020). On fast adversarial robustness adaptation in model-agnostic meta-learning. In *International Conference on Learning Representations*.

Wang, Z., Shen, Y., and Zavlanos, M. (2022). Risk-averse no-regret learning in online convex games. In *International Conference on Machine Learning*, pages 22999–23017. PMLR.

Wilder, B. (2018). Risk-sensitive submodular optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhang, J., Cheung, B., Finn, C., Levine, S., and Jayaraman, D. (2020). Cautious adaptation for reinforcement learning in safety-critical settings. In *International Conference on Machine Learning*, pages 11055–11065. PMLR.

Zhao, D., von Oswald, J., Kobayashi, S., Sacramento, J., and Grewe, B. F. (2020). Meta-learning via hypernetworks.

Contents

A Frequently Asked Question

Here we collected technical questions and suggestions from researchers who helped check out the manuscript. We thank these researchers for precious questions and provide more details.

Novelty & primary findings of this work. Here we mainly summarize two points of novelty in this work:

- *Meta Learning Robustification Framework.* Though the concept of the expected tail risk has emerged for several decades and has been widely employed in financial domains, the application to fast adaptation or robustification of meta learning remains limited in literature as far as we know.
- *Optimization Strategy & Theoretical Analysis.* We theoretically analyze two-stage optimization strategies as the heuristic algorithm in optimizing the distributionally robust meta learning models and demonstrate the improvement guarantee under certain conditions.

As verified in experimental results in the main paper, placing a probabilistic constraint in the task space is meaningful. It circumvents the effect of over-pessimistic consideration (worst-case optimization), increases robustness in proportional cases, and mostly retains or even improves average performance.

Apart from the novelty in the framework and algorithm parts, we have several findings which bring crucial insights into meta learning: (i) Not all tasks are necessary to perform fast adaptation. (ii) Additional focus on the tail risk has the potential to enhance models' generalization capability. (iii) The tail risk instead of extreme worst-case risk can better advance robustness in challenging datasets.

Meta risk function values as random variables. In some few-shot learning related work, the context and the target dataset are equivalently called the support and query datasets. The definition of a task in meta learning is up to application scenarios and specific meta learning algorithms or models. The commonly-used sinusoid regression using model-agnostic meta learning (Finn et al., 2017) considers the fixed number of context points to induce tasks. In contrast, conditional neural processes (Garnelo et al., 2018a) for few-shot regression vary the number of context points to induce tasks. Once the context and the target are partitioned and the model parameter is specified, we can obtain a meta risk function value. However, the meta risk function values are not in a compact Euclidean subspace in several cases.

The continuity of the meta risk probability density function. This relates to the task distribution and meta learning problems. Throughout the few-shot regression task, the meta risk function value can be approximately viewed as a continuous random variable. However, when it comes to the few-shot image classification mission, the meta risk function value is impossible to cover the entire continuous interval. In these scenarios, the probable values of accuracies are finite. This makes the theoretical analysis, *e.g.* Theorem (1), no longer holds. For example, Assumption (2) will be unrealistic when there exists a constant gap between two accuracy values. Hence, we leave this part a future research direction in theoretical analysis. Regarding the meta risk function, ours shares the same setup as that in (Fallah et al., 2021).

The selection of baselines in robust meta learning. A large body of prior work considers the robustness of meta learning in scenarios when the input of a data point, the model parameter, and the number of context points are trembled or modified. Robustness in presence of tasks distributions is seldom investigated except for the worst-case optimization in (Collins et al., 2020). Hence, we retain most of the setups in (Collins et al., 2020) for a fair comparison.

Influence of risk minimization principles. This paper is primarily devoted to studying the influence of the risk minimization principle on meta learning. The empirical risk minimization principle corresponds to reducing the Monte Carlo estimate of Average-case Meta Learning in this work. In comparison to TR-MAML (Collins et al., 2020), our approach has a couple of advantages as follows: (i) easier implementations. Note that min-max optimization is numerically unstable and requires a relaxation method for computationally intensive convex optimization, *e.g.*, robust stochastic mirror-prox algorithm used in TR-MAML (Collins et al., 2020). (ii) more flexible in terms of robustness concept. Theoretically, the worst-case meta learning corresponds to the extreme case of the distributionally robust risk minimization principle. (iii) empirically better performance in most cases. The expected tail risk minimization preserves a particular property that minimizing proportional worst-case fast adaptation seldom sacrifices the average performance. These advantages are also why we call our approach *simple yet effective*.

Comparison with Other Heuristic Optimization Strategies. To sum up the proposed optimization strategy, we empirically highlight the following points regarding the adopted heuristic strategies. There exist a couple of approximate algorithms for CVaR_α optimization. In comparison, the crude Monte Carlo one is the simplest for VaR_α estimates. It has an improvement guarantee under certain conditions, leaving it easier to analyze. Besides, we also compare ours to the risk reweighted method (Sagawa et al., 2020) in previously used benchmarks, and please take a closer look at that in Section (K).

B Pseudo Algorithms of DR-MAML & DR-CNPs

Algorithm 1: DR-MAML

Input : Task distribution $p(\tau)$; Confidence level α; Task batch size $\mathcal{B}$; Learning rates: λ_1 and λ_2.
Output : Meta-trained model parameter ϑ.
Randomly initialize the model parameter ϑ;
while *not converged* **do**
  Sample a batch of tasks $\{\tau_i\}_{i=1}^{\mathcal{B}} \sim p(\tau)$;
  // inner loop via gradient descent
  for $i = 1$ ***to*** $\mathcal{B}$ **do**
    Evaluate the gradient: $\nabla_\vartheta \ell(\mathfrak{D}_{\tau_i}^C; \vartheta)$ in Eq. (7);
    Perform task-specific gradient updates:
    $\vartheta_i \leftarrow \vartheta - \lambda_1 \nabla_\vartheta \ell(\mathfrak{D}_{\tau_i}^C; \vartheta)$;
  end
  // estimate $\text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)] \approx \hat{\xi}_\alpha$
  Evaluate performance $\mathcal{L}_{\mathcal{B}} = \{\ell(\mathfrak{D}_{\tau_i}^T; \vartheta_i)\}_{i=1}^{\mathcal{B}}$;
  Estimate $\text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)]$ and set $\xi = \hat{\xi}_\alpha$ in Eq. (7) with either percentile rank or density estimators;
  // outer loop via gradient descent
  Screen the subset $\mathcal{L}_{\hat{\mathcal{B}}} = \{\ell(\mathfrak{D}_{\hat{\tau}_i}^T; \vartheta_i)\}_{i=1}^{K}$ with $\hat{\xi}_\alpha$ for meta initialization updates;
  $\vartheta \leftarrow \vartheta - \lambda_2 \nabla_\vartheta \sum_{i=1}^{K} \ell(\mathfrak{D}_{\hat{\tau}_i}^T; \vartheta_i)$ in Eq. (7);
end

Algorithm 2: DR-CNP

Input : Task distribution $p(\tau)$; Confidence level α; Task batch size $\mathcal{B}$; Learning rate λ.
Output : Meta-trained model parameter ϑ.
Randomly initialize the model parameter ϑ;
while *not converged* **do**
  Sample a batch of tasks $\{\tau_i\}_{i=1}^{\mathcal{B}} \sim p(\tau)$;
  // estimate $\text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)]$
  Evaluate performance $\mathcal{L}_{\mathcal{B}} = \{\ell(\mathfrak{D}_{\tau_i}^T; z, \vartheta_i)\}_{i=1}^{\mathcal{B}}$;
  Estimate $\text{VaR}_\alpha[\ell(\mathcal{T}, \vartheta)] \approx \hat{\xi}_\alpha$ with either percentile rank or density estimators;
  // execute gradient descent
  Screen the subset $\mathcal{L}_{\hat{\mathcal{B}}} = \{\ell(\mathfrak{D}_{\hat{\tau}_i}^T; z, \vartheta)\}_{i=1}^{K}$ with $\hat{\xi}_\alpha$ for meta initialization updates;
  $\vartheta \leftarrow \vartheta - \lambda \nabla_\vartheta \sum_{i=1}^{K} \ell(\mathfrak{D}_{\hat{\tau}_i}^T; z, \vartheta)$ in Eq. (8);
end

C Properties of Risk Minimization Principles

C.1 Stochastic Optimization in the Constrained Form

In the section above, meta learning optimization objectives are discussed within three different principles, respectively the average-case in Eq. (1), the worst-case in Eq. (2) and CVaR_α worst-case in Eq. (5). This subsection continues this topic and introduces the relaxation variable ξ in the optimization objective. In this way, the robust fast adaptation can be reframed in the case of stochastic optimization with the probabilistic constraint.

We can equivalently express the minimization of the worst-case problem (Shalev-Shwartz and Wexler, 2016) within the following constrained stochastic optimization framework as follows.

$$\begin{aligned} &\min_{\vartheta\in\Theta,\xi\in\mathbb{R}^+} \xi \\ \text{s.t. } \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \leq \xi \quad &\text{with } \tau \in \mathcal{T} \end{aligned} \tag{9}$$

C.2 SGD Intractability of Meta Learning CVaR_α Optimization Objective

This subsection shows that directly stochastic gradient descent is intractable for meta learning to optimize CVaR_α. Note that the normalized density distribution function $p_\alpha(\tau;\vartheta)$ implicitly depends on ϑ and α, so we cannot access the exact form of such a distribution.

$$\nabla_\vartheta \mathcal{E}_\alpha(\vartheta) = \int p_\alpha(\tau;\vartheta)\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\nabla_\vartheta \ln p_\alpha(\tau;\vartheta) + \nabla_\vartheta \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\Big] d\tau \tag{10a}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K}\Bigg[\underbrace{\ell(\mathfrak{D}_{\tau_k}^T, \mathfrak{D}_{\tau_k}^C; \vartheta)\nabla_\vartheta \ln p_\alpha(\tau_k;\vartheta)}_{\text{Score Function}} + \nabla_\vartheta \ell(\mathfrak{D}_{\tau_k}^T, \mathfrak{D}_{\tau_k}^C; \vartheta)\Bigg] \tag{10b}$$

As illustrated in Eq. (10), the stochastic gradient estimate is not plausible since $p_\alpha(\tau;\vartheta)$ has no closed form. Hence, heuristic or convex programming algorithms for specific cases are mostly used as this domain's optimization strategy. However, designing optimization strategies for non-convex cases is non-trivial in this domain and requires more consideration of theoretical guarantees.

C.3 Risk-Sensitive Applications & Optimization Strategies

Related Applications. There are a number of applications concerning robust optimization with probabilistic constraints. Most of them are for the sake of safety. The risk principle CVaR_α firstly occurs in the financial domain as a coherent principle (Rockafellar et al., 2000). It enjoys much popularity in portfolio optimization (Quaranta and Zaffaroni, 2008). Gagne and Dayan (2021) adopt CVaR_α principles to improve the distributional reinforcement learning performance. To robustify robotic control, Pinto et al. (2017) varies hyper-parameters of Markov decision processes and optimizes proportional worst-case trajectories in policy optimization within the principle of CVaR_α. In work (Wilder, 2018), a CVaR_α related strategy is devised to solve submodular optimization problems. Such a risk measure is also included in producing robust options (Hiraoka et al., 2019). *Regarding probabilistic robust meta learning within CVaR_α principles, there exists scarce related work until now.*

Optimization Strategies. Concerning the constrained stochastic optimization problem, we particularly overview related work in this subsection. In the past few decades, there emerge substantial optimization strategies together with theoretical analysis for convex risk functions in CVaR_α optimization (Nemirovski et al., 2009; Fan et al., 2017; Meng et al., 2020; Levy et al., 2020; Duchi and Namkoong, 2021; Wang et al., 2022). As for the min-max risk minimization principle, which focuses on the worst case instead of a propositional worst cases, some researchers have designed relaxation or other heuristic algorithms to handle convex or non-convex risk functions (Chen et al., 2017; Collins et al., 2020; Jiang et al., 2021; Hsieh et al., 2021). Nevertheless, it remains challenging to design algorithms with convergence guarantees for non-convex risk function cases. One of the latest CVaR_α work on non-convex risk functions is (Sagawa et al., 2020), where a risk reweighted algorithm for robust neural networks is proposed to handle distributional shifts. Moreover, most CVaR_α optimization in non-convex risk functions follows this type of risk reweighted strategies in applications.

Remarks on Literature Work: Risk functions for robust fast adaptation cases are mostly non-convex, and it is non-trivial to design optimization strategies with a convergence guarantee. Meanwhile, we also conduct comparison experiments between work in (Sagawa et al., 2020) and ours in meta learning downstream tasks. We refer the reader to Appendix (K) for more details and analysis.

D Computational Complexity

The primary difference between distributionally robust meta learning and expected risk based meta learning lies in that a fixed probabilistic portion of task gradients are considered in meta updates. Such an operation drives the optimization procedure to focus more on proportional vulnerable scenarios, increasing the robustness in worst cases.

However, the iteration of the surrogate function $\varphi(\hat{\xi}_{\alpha_t};\vartheta)$ involves the screening of proportional worst cases since our strategies require extra evaluation of fast adaptation. This brings additional computational cost with complexity $\mathcal{O}\Big(\mathcal{B}\log(\mathcal{B})\Big)$, where $\mathcal{B}$ is the number of tasks in each batch. On the other hand, the proposed strategy performs sub-gradient updates instead of complete gradient updates. This helps reduce computational cost with complexity $\mathcal{O}\Big(\alpha\mathcal{B}|\mathcal{M}|\Big)$ in each iteration, where $|\mathcal{M}|$ corresponds to the scale of parameters of meta learning models and α is the confidence level in CVaR_α optimization.

Universally analyzing the computational complexity in meta learning is intractable since various meta learning methods exist. Some are gradient-based ones, while some are non-parametric ones. The exact number of iterations for convergence heavily relies on specific methods. In summary, given the same number of iterations and a fixed confidence level, the computational complexity difference for CVaR_α optimization in meta learning scenarios is $\mathcal{O}\Big(\mathcal{B}(\alpha|\mathcal{M}|-\log(\mathcal{B}))\Big)$.

To deepen understanding of our method, we explain more via specific examples. The main idea is to execute the sub-gradient operation over the batch of task gradients. Here we take the DR-MAML in Example (1) to show the operation and how the distributionally robust meta initialization is obtained:

$$\begin{aligned}\vartheta_{t+1}^{\text{meta}} \leftarrow \vartheta_t^{\text{meta}} - \lambda_1\Big[\sum_{i=1}^{\mathcal{B}}\nabla_\vartheta[\delta(\tau_i)\cdot\ell(\mathfrak{D}_{\tau_i}^T;\vartheta_t^{\tau_i})]\Big], \\ \text{with } \vartheta_t^{\tau_i} = \vartheta_t^{\text{meta}} - \lambda_2\nabla_\vartheta\ell(\mathfrak{D}_{\tau_i}^C;\vartheta),\ \tau_i\sim p(\tau),\end{aligned} \tag{11}$$

where λ_1 is the outer loop learning rate, λ_2 is the inner loop learning rate, and $\delta(\tau_i)$ is the indicator variable. Here $\delta(\tau_i)=1$ in the case when the $\ell(\mathfrak{D}_{\tau_i}^T;\vartheta_t^{\tau_i})$ falls into the $(1-\alpha)$-probabilistic worst-case region otherwise $\delta(\tau_i)=0$.

As for the optimal rate for convergence or the generalization bound, it is up to specific meta learning methods and risk minimization principles. For worst-case risk minimization for meta learning, there already exists theoretical analysis in previous work, *e.g.*, optimization-based meta learning (Collins et al., 2020) when fast adaptation functions hold the convexity property. When it comes to more universal cases, considering diverse meta learning methods and optimization strategies, it is still challenging to estimate the optimal rate for convergence. Also, our considered scenarios exist distributional drift between meta training and meta testing task distributions, which makes it tough to derive the generalization bound.

Finally, note that our developed optimization strategies for meta learning are regardless of meta learning methods, and DR-MAMLs and DR-CNPs are merely two examples. For the sake of convenience, we only implement DR-MAML to compare with TR-MAML in this work.

E Proof of Remark (2)

Remark (2). *If $\ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)$ is convex w.r.t. ϑ, then Eq.s (5)/(6) are also convex functions. In this case, the optimization objective Eq. (6) of our interest can be resolved with the help of convex programming algorithms (Fan et al., 2017; Meng et al., 2020; Levy et al., 2020).*

Proof: We at first show that $\left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi\right]^+ := \max\{\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi, 0\}$ is convex *w.r.t* ϑ if $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ is convex *w.r.t.* ϑ: For ease of derivation, let us redenote two functions $f_1(\xi; \vartheta) := \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi$ and $f_2(\xi; \vartheta) := 0$.

With any constant $\lambda \in [0, 1]$ and any two parameters $\vartheta_1 \in \Theta$ and $\vartheta_2 \in \Theta$, we can have:

$$\begin{aligned}
\left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \lambda\vartheta_1 + (1-\lambda)\vartheta_2) - \xi\right]^+ &= f_i(\xi; \lambda\vartheta_1 + (1-\lambda)\vartheta_2) \text{ [for some } i \in \{1, 2\}] && \text{(12a)}\\
&\leq \lambda f_i(\xi; \vartheta_1) + (1-\lambda) f_i(\xi; \vartheta_2) && \text{(12b)}\\
&\leq \lambda \max\{f_i(\xi; \vartheta_1)\}_{i=1,2} + (1-\lambda) \max\{f_i(\xi; \vartheta_2)\}_{i=1,2} && \text{(12c)}\\
&\leq \lambda \left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta_1) - \xi\right]^+ + (1-\lambda) \left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta_2) - \xi\right]^+, && \text{(12d)}
\end{aligned}$$

which shows that the risk function with slack variables is convex *w.r.t.* ϑ.

As $\varphi_\alpha(\xi; \vartheta) = \xi + \frac{1}{1-\alpha}\mathbb{E}_{p(\tau)}\left[\left[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi\right]^+\right]$ is the convex combination of the above mentioned convex function, the resulting $\varphi_\alpha(\xi; \vartheta)$ is naturally convex *w.r.t.* ϑ.

F Proof of Proposition (1)

Assumption (1): *The meta risk function $\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)$ is β_τ-Lipschitz continuous w.r.t. ϑ, which suggests: there exists a positive constant β_τ such that $\forall\{\vartheta, \vartheta'\} \in \Theta$:*

$$|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')| \leq \beta_\tau ||\vartheta - \vartheta'||.$$

Assumption (2): *For meta risk function values, the risk cumulative distribution function $F_\ell(l; \vartheta)$ is β_ℓ-Lipschitz continuous w.r.t. l, and the implicit normalized probability density function of tasks $p_\alpha(\tau; \vartheta)$ is β_θ-Lipschitz continuous w.r.t. ϑ.*

Assumption (3): *For any valid $\vartheta \in \Theta$ and corresponding implicit normalized probability density function of tasks $p_\alpha(\tau; \vartheta)$, the meta risk function value can be bounded by a positive constant $\mathcal{L}_{\max}$:*

$$\sup_{\tau \in \Omega_\tau^\alpha} \ell(\mathfrak{D}_{\tau_i}^T, \mathfrak{D}_{\tau_i}^C; \vartheta) \leq \mathcal{L}_{\max}.$$

Proposition (1): *Under Assumptions (1)/(2)/(3), the meta learning optimization objective $\mathcal{E}_\alpha(\vartheta)$ in Eq. (3) is continuous w.r.t. ϑ.*

Proof: Suppose that $\forall\{\vartheta, \vartheta'\} \in \Theta$ and $||\vartheta - \vartheta'|| < \delta$, we can have the following inequality based on Assumption (2):

$$\left|p_\alpha(\tau; \vartheta) - p_\alpha(\tau; \vartheta')\right| \leq \beta_\theta ||\vartheta - \vartheta'||.$$

Meanwhile, we can have the following inequality based on Assumption (1):

$$\left|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')\right| \leq \beta_\tau ||\vartheta - \vartheta'||.$$

Together with the boundness Assumption (3) of meta risk function value:

$$\sup_{\tau \in \Omega_\tau^\alpha} \ell(\mathfrak{D}_{\tau_i}^T, \mathfrak{D}_{\tau_i}^C; \vartheta) \leq \mathcal{L}_{\max},$$

we can roughly estimate the probabilistic constrained expected meta risk function values as follows:

$$\Big|\mathcal{E}_\alpha(\vartheta) - \mathcal{E}_\alpha(\vartheta')\Big| = \Big|\mathbb{E}_{p_\alpha(\tau;\vartheta)}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\Big] - \mathbb{E}_{p_\alpha(\tau;\vartheta')}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')\Big]\Big| \tag{13a}$$

$$= \Big|\mathbb{E}_{p_\alpha(\tau;\vartheta)}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\Big] - \mathbb{E}_{p_\alpha(\tau;\vartheta')}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\Big] \tag{13b}$$

$$+\mathbb{E}_{p_\alpha(\tau;\vartheta')}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')\Big]\Big| \tag{13c}$$

$$\leq \int \Big|p_\alpha(\tau;\vartheta) - p_\alpha(\tau;\vartheta')\Big|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)d\tau + \mathbb{E}_{p_\alpha(\tau;\vartheta')}\Big[\Big|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta')\Big|\Big] \tag{13d}$$

$$\leq \beta_\theta||\vartheta - \vartheta'|| \sup_{\tau\in\Omega_\tau^\alpha}\{\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)\} + \sup_{\tau\in\Omega_\tau^\alpha}\{\beta_\tau\}||\vartheta - \vartheta'|| \tag{13e}$$

$$= \Big(\beta_\theta\mathcal{L}_{\max} + \beta_{\max}\Big)||\vartheta - \vartheta'||. \tag{13f}$$

From the above inequality, we can see that $\mathcal{E}_\alpha(\vartheta) - \mathcal{E}_\alpha(\vartheta')$ is a $(\beta_\theta\mathcal{L}_{\max} + \beta_{\max})$–Lipschitz continuous *w.r.t.* ϑ. As a result, we demonstrate Proposition (1).

G Proof of Proposition (2)

Proposition (2): *Suppose there exists $\delta \in \mathbb{R}^+$ such that $|\xi_\alpha(\vartheta) - \hat{\xi}_\alpha(\vartheta)| < \delta$ with $\hat{\xi}_\alpha(\vartheta)$ an estimate of $\xi_\alpha(\vartheta)$. Then there exists a constant $\kappa_\alpha = \max\{\frac{2-\alpha}{1-\alpha}, \frac{\alpha}{1-\alpha}\}$ such that*

$$\varphi_\alpha(\hat{\xi}_\alpha(\vartheta); \vartheta) - \kappa_\alpha\delta < \mathcal{E}_\alpha(\vartheta) \leq \varphi_\alpha(\hat{\xi}_\alpha(\vartheta); \vartheta).$$

Proof: Based on the direct deduction in work (Rockafellar et al., 2000), we know that for any $\vartheta \in \Theta$, the inequality holds: $\varphi_\alpha(\xi_\alpha; \vartheta) \leq \varphi_\alpha(\hat{\xi}_\alpha; \vartheta)$.

In the case when $\hat{\xi}_\alpha = \xi_\alpha + \delta$ with $\delta \in \mathbb{R}^+$, the probability space of the task Ω_τ can be respectively partitioned into three disjoint probability space:

$$\mathbb{P}^-(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell)|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \leq \xi_\alpha, \tau \in \Omega_\tau\}\Big) \tag{14a}$$

$$\mathbb{P}^+(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell)|\xi_\alpha < \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \leq \xi_\alpha + \delta, \tau \in \Omega_\tau\}\Big) \tag{14b}$$

$$\mathbb{P}^{++}(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell)|\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) > \xi_\alpha + \delta, \tau \in \Omega_\tau\}\Big). \tag{14c}$$

As a result, we can estimate the difference between the two terms as follows:

$$\varphi_\alpha(\hat{\xi}_\alpha; \vartheta) - \varphi_\alpha(\xi_\alpha; \vartheta) = \hat{\xi}_\alpha - \xi_\alpha \tag{15a}$$

$$+\frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi_\alpha\Big] + \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\xi_\alpha - \hat{\xi}_\alpha\Big] \tag{15b}$$

$$\leq \delta + \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\delta\Big] - \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\delta\Big] \leq \delta + \frac{1}{1-\alpha}\delta = \frac{2-\alpha}{1-\alpha}\delta. \tag{15c}$$

Similarly, in the case when $\hat{\xi}_\alpha = \xi_\alpha - \delta$ with $\delta \in \mathbb{R}^+$, the probability space of the task Ω_τ can be partitioned into three disjoint space with the probability respectively:

$$\mathbb{P}^{--}(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell) | \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \leq \xi_\alpha - \delta, \tau \in \Omega_\tau\}\Big) \tag{16a}$$

$$\mathbb{P}^{-}(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell) | -\delta < \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \xi_\alpha < 0, \tau \in \Omega_\tau\}\Big) \tag{16b}$$

$$\mathbb{P}^{+}(\tau) = \mathbb{P}\Big(\{\mathcal{M}_\vartheta^{-1}(\ell) | \ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) \geq \xi_\alpha, \tau \in \Omega_\tau\}\Big). \tag{16c}$$

As a result, we can estimate the difference between the two terms as follows:

$$\varphi_\alpha(\hat{\xi}_\alpha; \vartheta) - \varphi_\alpha(\xi_\alpha; \vartheta) = \hat{\xi}_\alpha - \xi_\alpha \tag{17a}$$

$$+ \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta) - \hat{\xi}_\alpha\Big] + \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\xi_\alpha - \hat{\xi}_\alpha\Big] \tag{17b}$$

$$\leq -\delta + \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\delta\Big] + \frac{1}{1-\alpha}\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\delta\Big] \leq -\delta + \frac{1}{1-\alpha}\delta = \frac{\alpha}{1-\alpha}\delta. \tag{17c}$$

Based on the inequalities (15)/(17), we can have $\kappa_\alpha = \max\{\frac{2-\alpha}{1-\alpha}, \frac{\alpha}{1-\alpha}\}$ such that the proposition is verified as $\varphi_\alpha(\hat{\xi}_\alpha; \vartheta) - \kappa_\alpha\delta < \varphi_\alpha(\xi_\alpha; \vartheta)$.

H Proof of Improvement Guarantee in Theorem (1)

Theorem (1): *Under assumptions (1)/(2)/(3), suppose that the estimate error with the crude Monte Carlo holds:* $|\hat{\xi}_{\alpha_t} - \xi_{\alpha_t}| \leq \frac{\lambda}{\beta_\ell(1-\alpha)^2}, \forall t \in \mathbb{N}^+$, *with the subscript* t *the iteration number,* λ *the learning rate in stochastic gradient descent,* β_ℓ *the Lipschitz constant of the risk cumulative distribution function, and* α *the confidence level. Then the proposed heuristic algorithm with the crude Monte Carlo can produce at least a local optimum for distributionally robust fast adaptation.*

Proof: In the main paper, Fig. (2) provides the outline of the improvement guarantee proof. Note that $\hat{\xi}_{\alpha_t}$ is an estimate of ξ_{α_t} with the help of Monte Carlo samples, and this depends on the model parameters ϑ_t in optimization. Performing the gradient updates *w.r.t.* the surrogate function $\varphi_\alpha(\hat{\xi}_\alpha; \vartheta)$, we can have the following equation with a small step-size learning rate λ:

$$\begin{aligned} &\text{Gradient Descent} : \vartheta_{t+1} = \vartheta_t - \lambda\nabla_\vartheta\varphi_\alpha(\hat{\xi}_\alpha; \vartheta) \\ \Rightarrow &\text{Monotonic Sequence} : \varphi_\alpha(\hat{\xi}_{\alpha_t}; \vartheta_{t+1}) \leq \varphi_\alpha(\hat{\xi}_{\alpha_t}; \vartheta_t). \end{aligned} \tag{18}$$

To verify the improvement guarantee, we need to show that with the meta model parameters derived from the surrogate function:

$$\varphi_\alpha(\xi_{\alpha_{t+1}}; \vartheta_{t+1}) \leq \varphi_\alpha(\xi_{\alpha_t}; \vartheta_t). \tag{19}$$

With the previous deduction $\varphi_\alpha(\xi_{\alpha_{t+1}}; \vartheta_{t+1}) \leq \varphi_\alpha(\xi_{\alpha_t}; \vartheta_{t+1})$ from the property of CVaR_α, the demonstration is equivalently reduced to show that:

$$\varphi_\alpha(\xi_{\alpha_t}; \vartheta_{t+1}) \leq \varphi_\alpha(\xi_{\alpha_t}; \vartheta_t). \tag{20}$$

We can perform the one order Taylor expansion with Peano's form of remainders *w.r.t.* $\varphi_\alpha(\xi_{\alpha_t}; \vartheta)$ around the point ϑ_t:

$$\begin{aligned} \varphi_\alpha(\xi_{\alpha_t}; \vartheta_{t+1}) = \varphi_\alpha(\xi_{\alpha_t}; \vartheta_t) - \lambda\Big[\nabla_\vartheta\varphi_\alpha(\hat{\xi}_{\alpha_t}; \vartheta)|_{\vartheta=\vartheta_t}^T \cdot \nabla_\vartheta\varphi_\alpha(\xi_{\alpha_t}; \vartheta)|_{\vartheta=\vartheta_t}\Big] \\ + \mathcal{O}(|\vartheta_{t+1} - \vartheta_t|) \leq \varphi_\alpha(\xi_{\alpha_t}; \vartheta_t). \end{aligned} \tag{21}$$

In the case when $\hat{\xi}_\alpha = \xi_\alpha + \delta$ with $\delta \in \mathbb{R}^+$, we use the partitioned task probability space in Eq. (14) and can derive the gradient estimate:

$$\begin{aligned} \nabla_\vartheta\varphi_\alpha(\hat{\xi}_{\alpha_t}; \vartheta)|_{\vartheta=\vartheta_t}^T = \frac{1}{1-\alpha}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)|_{\vartheta=\vartheta_t}\Big]\Big] \\ + \frac{1}{1-\alpha}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C; \vartheta)|_{\vartheta=\vartheta_t}\Big]\Big]. \end{aligned} \tag{22}$$

With $|F_\ell(\hat{\xi}_\alpha;\vartheta) - F_\ell(\xi_\alpha;\vartheta)| \leq \beta_\ell \delta$ in the Assumption (2),

$$
\begin{aligned}
&-\mathbb{E}_{\tau\sim\mathbb{P}^{+}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big] &\text{(23a)}\\
&\leq ||\mathbb{E}_{\tau\sim\mathbb{P}^{+}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]|| \cdot ||\mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]|| &\text{(23b)}\\
&\leq \mathbb{E}_{\tau\sim\mathbb{P}^{+}(\tau)}\Big[\sup_{\tau\in\Omega_\tau} ||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\Big] \cdot \mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\sup_{\tau\in\Omega_\tau} ||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\Big] &\text{(23c)}\\
&\leq \beta_\ell\delta(1-\alpha)\left(\sup_{\tau\in\Omega_\tau} ||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\right)^2 = \beta_\ell\delta(1-\alpha)\mu^2, &\text{(23d)}
\end{aligned}
$$

where μ defines the $\sup_{\tau\in\Omega_\tau} ||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||$.

Then we can derive the following inequality with the deduction from Eq. (23):

$$
\begin{aligned}
\varphi_\alpha(\xi_{\alpha_t};\vartheta_{t+1}) &= \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) &\text{(24a)}\\
&-\lambda\Big[\nabla_\vartheta\varphi_\alpha(\hat{\xi}_{\alpha_t};\vartheta)|_{\vartheta=\vartheta_t}^T \cdot \nabla_\vartheta\varphi_\alpha(\xi_{\alpha_t};\vartheta)|_{\vartheta=\vartheta_t}\Big] + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) &\text{(24b)}\\
&= \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) &\text{(24c)}\\
&-\frac{\lambda}{(1-\alpha)^2}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\Big] &\text{(24d)}\\
&-\frac{\lambda}{(1-\alpha)^2}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^{+}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\Big] &\text{(24e)}\\
&+\mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) &\text{(24f)}\\
&\leq \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) - \frac{\lambda||\nu_1||_2^2}{(1-\alpha)^2} + \beta_\ell\delta(1-\alpha)\mu^2 + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|), &\text{(24g)}
\end{aligned}
$$

where $\nu_1 = \mathbb{E}_{\tau\sim\mathbb{P}^{++}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]$.

To ensure the existence of improvement guarantee, we need that the following inequality holds:

$$
\varphi_\alpha(\xi_{\alpha_t};\vartheta_t) - \frac{\lambda||\nu_1||_2^2}{(1-\alpha)^2} + \beta_\ell\delta(1-\alpha)\mu^2 + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) \leq \varphi_\alpha(\xi_{\alpha_t};\vartheta_t). \tag{25}
$$

Similarly, in the case when $\hat{\xi}_\alpha = \xi_\alpha - \delta$ with $\delta \in \mathbb{R}^+$, we use the probability space of partitioned tasks in Eq. (16) and can derive the gradient estimate:

$$
\begin{aligned}
\nabla_\vartheta\varphi_\alpha(\hat{\xi}_{\alpha_t};\vartheta)|_{\vartheta=\vartheta_t}^T = \frac{1}{1-\alpha}\Big[&\mathbb{E}_{\tau\sim\mathbb{P}^{-}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\\
&+\mathbb{E}_{\tau\sim\mathbb{P}^{+}(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\Big].
\end{aligned} \tag{26}
$$

With $|F_\ell(\hat{\xi}_\alpha;\vartheta) - F_\ell(\xi_\alpha;\vartheta)| \le \beta_\ell \delta$ in the Assumption (2), the following formula can be easily verified:

$$-\mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big] \tag{27a}$$

$$\le ||\mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]|| \cdot ||\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]|| \tag{27b}$$

$$\le \mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\sup_{\tau\in\Omega_\tau}||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\Big] \cdot \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\sup_{\tau\in\Omega_\tau}||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\Big] \tag{27c}$$

$$\le \beta_\ell\delta(1-\alpha)\left(\sup_{\tau\in\Omega_\tau}||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\right)^2 = \beta_\ell\delta(1-\alpha)\mu^2. \tag{27d}$$

Once again, we can derive the following inequality with the deduction from Eq. (27):

$$\varphi_\alpha(\xi_{\alpha_t};\vartheta_{t+1}) = \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) - \lambda\Big[\nabla_\vartheta\varphi_\alpha(\hat{\xi}_{\alpha_t};\vartheta)|_{\vartheta=\vartheta_t}^T \cdot \nabla_\vartheta\varphi_\alpha(\xi_{\alpha_t};\vartheta)|_{\vartheta=\vartheta_t}\Big] + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) \tag{28a}$$

$$= \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) \tag{28b}$$

$$-\frac{\lambda}{(1-\alpha)^2}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\Big] \tag{28c}$$

$$-\frac{\lambda}{(1-\alpha)^2}\Big[\mathbb{E}_{\tau\sim\mathbb{P}^-(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]^T \cdot \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]\Big] \tag{28d}$$

$$+\mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) \tag{28e}$$

$$\le \varphi_\alpha(\xi_{\alpha_t};\vartheta_t) - \frac{\lambda||\nu_2||_2^2}{(1-\alpha)^2} + \beta_\ell\delta(1-\alpha)\mu^2 + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|), \tag{28f}$$

where $\nu_2 = \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\Big]$.

To ensure the existence of improvement guarantee, we need that the following holds:

$$\varphi_\alpha(\xi_{\alpha_t};\vartheta_t) - \frac{\lambda||\nu_2||_2^2}{(1-\alpha)^2} + \beta_\ell\delta(1-\alpha)\mu^2 + \mathcal{O}(|\vartheta_{t+1}-\vartheta_t|) \le \varphi_\alpha(\xi_{\alpha_t};\vartheta_t). \tag{29}$$

Considering the above two cases and estimated bounds Eq. (25)/(29), we can roughly estimate the upper bound of the required δ to guarantee performance improvement using our developed strategy in optimization:

$$\delta \le \frac{\lambda||\nu_{\mathrm{m}}||_2^2}{\beta_\ell(1-\alpha)^3\mu^2}, \tag{30}$$

where λ is the formerly mentioned learning rate, $||\nu_{\mathrm{m}}||_2^2$ is $\max\{||\nu_1||_2^2, ||\nu_2||_2^2\}$, and μ is supermum of the meta risk function derivatives in the task domain $\sup_{\tau\in\Omega_\tau}||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||$.

With the help of Jensen inequality, $||\nu_i||_2^2$ can be roughly bounded as:

$$\begin{aligned}||\nu_i||_2^2 &\le \mathbb{E}_{\tau\sim\mathbb{P}^+(\tau)}\Big[\big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\big]^T \cdot \big[\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}\big]\Big] \\ &\le (1-\alpha)\left(\sup_{\tau\in\Omega_\tau}||\nabla_\vartheta \ell(\mathfrak{D}_\tau^T,\mathfrak{D}_\tau^C;\vartheta)|_{\vartheta=\vartheta_t}||\right)^2 = (1-\alpha)\mu^2.\end{aligned} \tag{31}$$

With Eq.s (30)/(31), we can finally obtain the necessary condition for improvement guarantee:

$$\delta \leq \frac{\lambda}{\beta_\ell (1-\alpha)^2}. \tag{32}$$

I Proof of Approximation Error in Theorem (2)

This section is to build up connections between the number of Monte Carlo samples in estimating VaR_α and the gap of solutions between the approximately derived one and the theoretical one.

Theorem 2 (Gaps of Optimized Solutions) *Suppose $F_\ell(l;\vartheta) \in \mathcal{C}^2$ in l-domain. With meta trained ϑ_*, the crude Monte Carlo estimate of $\hat{\xi}_\alpha$, the constant κ_α in Proposition (2), and the sufficiently large number of the task batch $\mathcal{B}$, and $\mathcal{R}_\mathcal{B} = \mathcal{O}(\mathcal{B}^{-3/4} \ln \mathcal{B})$, we have the expected error between the exact optimum and the approximate optimum:*

$$\mathcal{E}_\alpha(\vartheta_*) \leq \varphi_\alpha(\hat{\xi}_\alpha;\vartheta_*) + \kappa_\alpha \left[\frac{\alpha - \hat{F}_\ell(\hat{\xi}_\alpha;\mathcal{B},\vartheta_*)}{\frac{dF_\ell(\xi;\vartheta)}{d\xi}|_{\xi=\xi_\alpha}} + \mathcal{R}_\mathcal{B} \right].$$

Proof: As noted in (Bahadur, 1966), with assumptions that the cumulative distribution function $F_\ell(l;\vartheta)$'s second order derivative is continuous in l-domain, namely $F_\ell(l;\vartheta) \in \mathcal{C}^2$, and $\frac{dF_\ell(\xi;\vartheta)}{d\xi}|_{\xi=\xi_\alpha}$, the quantile estimate with crude Monte Carlo can be asymptotically written in the form with the help of central limit theory (Rosenblatt, 1956):

$$\hat{\xi}_\alpha - \xi_\alpha = \frac{\alpha - \hat{F}_\ell(\hat{\xi}_\alpha;\mathcal{B},\vartheta_*)}{\frac{dF_\ell(\xi;\vartheta_*)}{d\xi}|_{\xi=\xi_\alpha}} + \mathcal{R}_\mathcal{B}, \quad \text{with } \mathcal{R}_\mathcal{B} = \mathcal{O}(\mathcal{B}^{-3/4}\ln\mathcal{B}) \text{ when } \mathcal{B} \to \infty, \tag{33}$$

where the empirical cumulative distribution function is computed as follows.

With the sampled meta risk function values $\mathcal{L}_\mathcal{B} = \{\ell(\mathfrak{D}^T_{\tau_i}, \mathfrak{D}^C_{\tau_i};\vartheta)\}_{i=1}^{\mathcal{B}}$, we rank them by values as $\hat{\mathcal{L}}_\mathcal{B} = \{\ell(\mathfrak{D}^T_{\hat{\tau}_i}, \mathfrak{D}^C_{\hat{\tau}_i};\vartheta)\}_{i=1}^{\mathcal{B}}$, which means $\ell(\mathfrak{D}^T_{\hat{\tau}_{i-1}}, \mathfrak{D}^C_{\hat{\tau}_{i-1}};\vartheta) \leq \ell(\mathfrak{D}^T_{\hat{\tau}_i}, \mathfrak{D}^C_{\hat{\tau}_i};\vartheta)$. Then the empirical cumulative distribution function with these order statistics (Barabás, 1987) can be written as:

$$\hat{F}_\ell(\xi;\mathcal{B},\vartheta) = \begin{cases} 0, & \text{if } \xi \leq \ell(\mathfrak{D}^T_{\hat{\tau}_1}, \mathfrak{D}^C_{\hat{\tau}_1};\vartheta) \\ \frac{k}{\mathcal{B}}, & \text{if } \ell(\mathfrak{D}^T_{\hat{\tau}_k}, \mathfrak{D}^C_{\hat{\tau}_k};\vartheta) < \xi \leq \ell(\mathfrak{D}^T_{\hat{\tau}_{k+1}}, \mathfrak{D}^C_{\hat{\tau}_{k+1}};\vartheta) \quad (k = 1, 2, \dots, \mathcal{B}) \\ 1, & \text{if } \ell(\mathfrak{D}^T_{\hat{\tau}_\mathcal{B}}, \mathfrak{D}^C_{\hat{\tau}_\mathcal{B}};\vartheta) < \xi. \end{cases} \tag{34}$$

Based on Proposition (2) and $\kappa_\alpha = \max\{\frac{2-\alpha}{1-\alpha}, \frac{\alpha}{1-\alpha}\}$, we know the inequality holds:

$$\varphi_\alpha(\hat{\xi}_\alpha(\vartheta);\vartheta) - \mathcal{E}_\alpha(\vartheta) < \kappa_\alpha \delta. \tag{35}$$

With Eq. (33)/(35), the following inequality naturally holds when $\mathcal{B}$ is large enough:

$$\varphi_\alpha(\hat{\xi}_\alpha;\vartheta_*) \leq \mathcal{E}_\alpha(\vartheta_*) + \kappa_\alpha \left[\frac{\alpha - \hat{F}_\ell(\xi_\alpha;\mathcal{B},\vartheta_*)}{\frac{dF_\ell(\xi;\vartheta)}{d\xi}|_{\xi=\xi_\alpha}} + \mathcal{R}_\mathcal{B} \right].$$

J Experimental Set-up & Implementation Details

This section is to provide experimental details in this paper. For the implementation of few-shot sinusoid regression and few-shot image classification, we respectively refer the reader to TR-MAML's codes (`https://github.com/lgcollins/tr-maml`) in (Collins et al., 2020) and vanilla MAML's codes (`https://github.com/AntreasAntoniou/HowToTrainYourMAMLPytorch`) in (Antoniou et al., 2019). And ours is built on top of the above codes except for simple modification of loss functions. The learning rates for the inner loop and the outer loop of all methods are the same as the above ones.

To facilitate the use of our heuristic optimization strategy, we leave the pytorch version of loss functions within the expected tail risk minimization. The example is provided in the case of mean square errors after MAML's inner loop, which is *simple to implement yet effective in robustifying fast adaptation*, as follows:

```python
import torch
from torch.nn import MSELoss

def cvar_mse(y_pred,y_true,conf_level=0.5):

    batch_MSE=MSELoss(reduction='none')
    batch_loss=batch_MSE(y_pred,y_true)

    # average risk values over non-task dimensions
    batch_avg_loss=torch.mean(batch_loss,dim=-1)

    # crude Monte Carlo to estimate VaR and sub-tasks
    topk_mse,topk_idxs=torch.topk(batch_avg_loss,int((1-conf_level)*
    y_true.size()[0]))

    return torch.mean(topk_mse)
```

Listing 1: Loss Functions in Two-Stage Heuristic Algorithm with Crude Monte Carlo

J.1 Meta Learning Datasets & Tasks

Sinusoid Regression. MAML (Finn et al., 2017), TR-MAML (Collins et al., 2020), and DR-MAML are considered in this experiment. We retain the setup in task generation and partition in (Collins et al., 2020). More specifically, there exists a distribution drift between the meta-training and the meta-testing function families $\{f_m(x) = a_m \sin(x - b_m)\}_{m=1}^{M}$.

Numerous easy tasks and a small proportion of difficult tasks are available in meta-training, while all tasks in the space are used in the evaluation. The range of the phase parameter is $b \in [0, \pi]$, and the amplitude range of the parameter is $a \in [0.1, 1.05]$ for easy tasks and $a \in [4.95, 5.0]$ for difficult tasks. It is noted that the sinusoid task is more challenging to fit with larger amplitudes as the resulting function is less smooth. The loss function corresponds to the mean-squared error between the predicted value $f(x)$ and the ground truth value. The number of task batches is 50 for 5-shot and 25 for 10-shot. The optimal selection of the confidence level α is difficult since we need to trade off the worst and average performance. Our setup is to minimize CVaR_α, which already considers the worst-case at some degree, so we watch the average performance in meta training results and set $\alpha = 0.7$ for all few-shot regression tasks. There is no external configuration for this hyper-parameter. The maximum number of iterations in meta-training is 70000.

Few-shot Image Classification. MAML (Finn et al., 2017), TR-MAML (Collins et al., 2020), and DR-MAML are considered in this experiment. The `N-way K-shot` classification corresponds to an `N`-classification problem with `K`-labeled examples available to the meta learner.

The Omniglot dataset consists of 1623 handwritten characters from 50 alphabets, with each 20 examples. The task distribution is uniform for all task instances consisting of characters from one specific alphabet. The dataset split follows procedures in (Triantafillou et al., 2019). Finally, 25 alphabets are used for meta-training, with 20 alphabets for meta-testing. The number of task batches is 16. The confidence level $\alpha = 0.5$ is selected with the same criteria as that in few-shot regression tasks. The maximum number of iterations is 60000 in meta-training. As the construction of the Omniglot meta dataset is related to specific alphabets and the scale of combination for tasks is huge, this indicates randomly sampled meta-training tasks in the evaluation of the main paper Tables may not be used in meta-training.

The *mini*-ImageNet dataset is pre-processed according to (Larochelle, 2017). In detail, 64 classes are used for meta-training, with the remaining 36 classes for meta-testing. Tasks are generated as follows: 64 meta-training classes are randomly grouped into 8 meta-train tasks with the class numbers $\{6, 7, 7, 8, 8, 9, 9, 10\}$, and the 36 meta-testing classes are processed in the same way. Finally, each task is built by sampling 1 image from 5 different classes within one task, resulting in a `5-way 1-shot` problem. The number of task batches is 4. The maximum number of iterations is 60000 in meta-training.

J.2 Neural Architectures & Optimization

Sinusoid Regression. MAML (Finn et al., 2017), TR-MAML (Collins et al., 2020), and DR-MAML are considered in this experiment. We retain the neural architecture for regression problems in (Finn et al., 2017; Collins et al., 2020). That is, we deploy a fully-connected neural network with two hidden layers of 40 ReLU nonlinear activation units. All methods use one stochastic gradient descent step as the inner loop.

Few-shot Image Classification. We retain the neural architecture for few-shot image classification problems in (Finn et al., 2017; Collins et al., 2020). In detail, a four-layer convolutional neural network is used for both Omniglot and *mini*-ImageNet datasets. All methods use one stochastic gradient descent step as the inner loop.

K Additional Experimental Results

More Quantitative Analysis. Due to the page limit in the main paper, we include the α's sensitivity experimental result in sinusoid `10-shot` regression. As illustrated in Fig. (7), the trend is similar to that in sinusoid `5-shot` regression. Worst-case optimization degrades the average performance of TR-MAML. DR-MAML is entangled with MAML in the average performance, while the performance gap between them is significant in the worst-case.

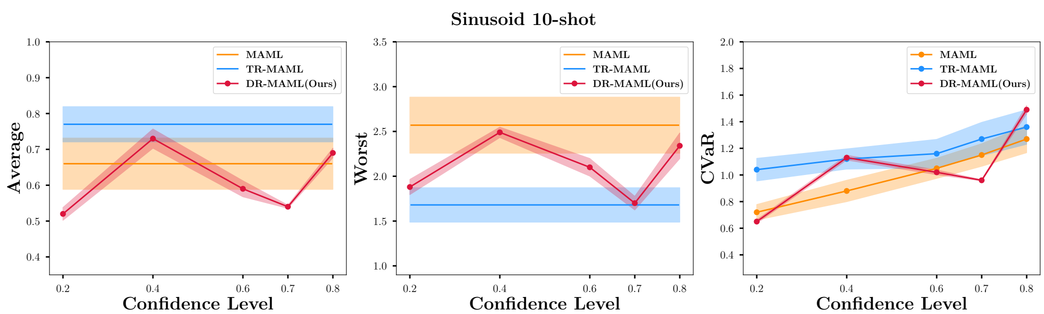

Figure 7: **Meta Testing Performance of Meta-Trained DR-MAML with Various Confidence Levels α.** MAML and TR-MAML are irrelevant with the variation of α in meta-training. The plots report testing MSEs with standard error bars in shadow regions.

Regarding few-shot image classification in the *mini*-ImageNet dataset, we can observe that in Fig. (8), the standard error is relatively smaller than in previous regression cases. When the confidence level is over a particular value, *e.g.* $\alpha > 0.5$, there occurs a significant decline of performance in all metrics. Note that when $\alpha \to 1.0$, the optimization objective approaches the worst-case optimization objective. Here we attach two possible reasons for the performance degradation phenomenon: (i) The adopted base optimization technique matters in nearly worst-case optimization. The stochastic mirror descent-ascent (Juditsky et al., 2011) is utilized in TR-MAML, which is more stable in deriving the optimal solution. In comparison, the stochastic gradient descent with sub-gradient operations works as the optimization method, and this method can be unstable when the scale of worst-case examples is small in the update. (ii) For few-shot image classification, estimates of VaR_α can be less precise with limited batch sizes and higher α values since the meta risk function value is discontinuous. Consequently, we can also attribute the severe degradation of fast adaptation performance in higher α value cases to the approximation errors of quantile estimates.

More Visualization Results. Further, we explore the influence of the expected tail risk minimization principle in meta learning. Here the landscape of meta risk values, namely fast adaptation losses, is presented in the sinusoid regression problem.

As exhibited in Fig. (9), the evaluated meta risk values from one random trial are associated with hyper-parameters of tasks. The final optimized results can discover some tasks difficult in fast adaptations. Meta learning methods are difficult to adapt in task regions with higher amplitudes. MAML exhibits higher MSEs in regions with the amplitude $a > 2$, while DR-MAML minimizes a proportion of risks in these regions. In contrast, TR-MAML reduces the risk around task regions with $a > 2$ to a certain extent; however, it shows relatively higher risk values in easy regions. Such

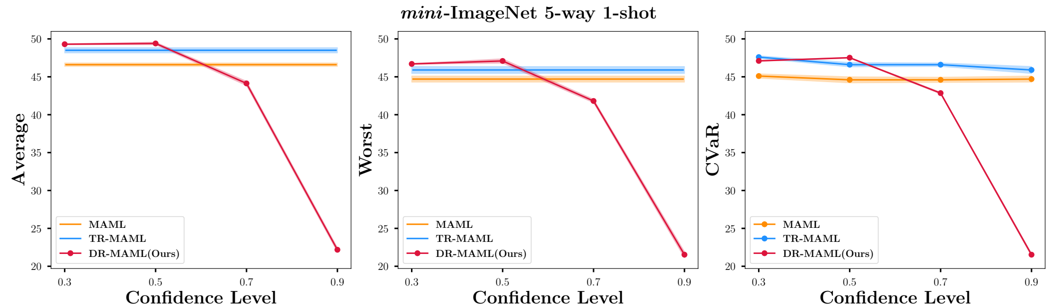

Figure 8: **Meta Testing Classification Accuracies of Meta-Trained DR-MAML with Various Confidence Levels** α**.** MAML and TR-MAML are irrelevant with the variation of α in meta-training. The plots report testing accuracies with standard error bars in shadow regions.

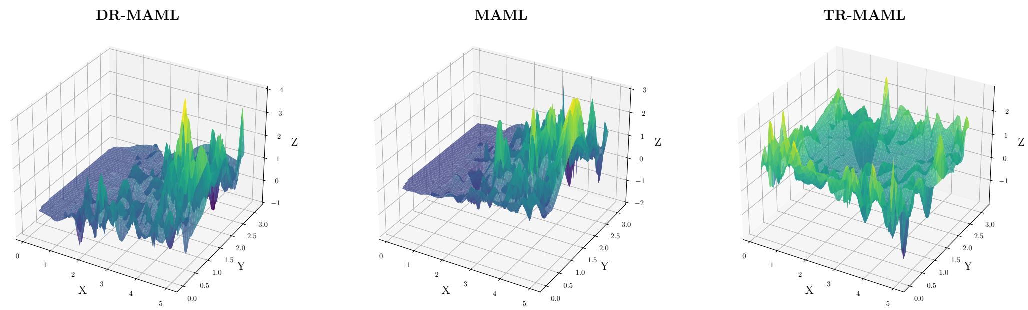

Figure 9: **The Fast Adaptation Risk Landscape of Meta-Trained DR-MAML, TR-MAML and MAML.** Shown is an example of sinusoid `5-shot` regression, which corresponds to the function space $f(x) = a\sin(x - b)$. The X-axis denotes the amplitude parameter a, and the Y-axis is the phase parameter b. The confidence level is $\alpha = 0.7$ in meta-training. The plots report testing MSEs in the Z-axis with a random trial of generating tasks.

evidence reflects the interpretability in optimization within the expected tail risk minimization, and the landscape of meta risk values is relatively flat and smooth than others.

Comparison with Other Optimization Strategies. Note that instantiations of distributionally robust meta learning methods, such as DR-MAML and DR-CNPs in Examples (1)/(2) are regardless of optimization strategies and can be optimized via any heuristic algorithms for CVaR_α objectives.

Additionally, we use DR-MAML as the example and perform the comparison between our two-stage algorithm and the risk reweighted algorithm (Sagawa et al., 2020). The intuition of the risk reweighted algorithm is to relax the weights of tasks and assign more weights to the gradient of worst cases. The normalization of risk weights is achieved via the softmax operator. Though there is an improvement guarantee *w.r.t.* the probabilistic worst group of tasks, the algorithm is not specially designed for meta learning or CVaR_α objective.

$$\min_{\vartheta\in\Theta} \mathcal{E}_\alpha(\vartheta) := \mathbb{E}_{p(\tau;\vartheta)}\Big[\frac{p_\alpha(\tau;\vartheta)}{p(\tau;\vartheta)}\ell(\mathfrak{D}_\tau^T, \mathfrak{D}_\tau^C;\vartheta)\Big] \approx \frac{1}{\mathcal{B}}\sum_{b=1}^{\mathcal{B}} \omega_b(\tau_b;\vartheta)\ell(\mathfrak{D}_{\tau_b}^T, \mathfrak{D}_{\tau_b}^C;\vartheta) \tag{36}$$

Meanwhile, the weight of task gradients after normalization is a biased estimator *w.r.t.* the constrained probability $p_\alpha(\tau;\vartheta)$ in the task space. In other words, the risk reweighted method can be viewed as approximation *w.r.t.* the importance weighted method in Eq. (36). In the importance weighted method, for tasks out of the region of $(1-\alpha)$-proportional worst, the probability of sampling such tasks τ_b is zero, indicating $\omega_b(\tau_b;\vartheta) = 0$. While in risk reweighted methods, the approximate weight is assumed to satisfy $\omega_b(\tau_b;\vartheta) \propto \exp\Big(\frac{\ell(\mathfrak{D}_{\tau_b}^T, \mathfrak{D}_{\tau_b}^C;\hat{\vartheta})}{\tau}\Big)$, where $\hat{\vartheta}$ means last time updated meta model parameters and the risk function value is evaluated after fast adaptation.

In implementations, we keep the setup the same as Risk-Rweighted methods in (Paszke et al., 2019) for meta-training. As illustrated in Table (4)/(5), DR-MAML with the two-stage optimization

Table 4: **Test average mean square errors (MSEs) with reported standard deviations for sinusoid regression (5 runs). We mainly compare DR-MAML with different optimization algorithms.** `5-shot` and `10-shot` cases are respectively considered here. The results are evaluated across the 490 meta-test tasks, which is the same as in (Collins et al., 2020). With $\alpha = 0.7$ for meta training, the best testing results are in bold.

Method	5-shot			10-shot		
	Average	Worst	CVaR_α	Average	Worst	CVaR_α
DR-MAML (Risk-Reweighted, (Sagawa et al., 2020))	0.91$\pm$0.06	3.57$\pm$0.56	1.83$\pm$0.03	0.61$\pm$0.02	1.90$\pm$0.11	1.13$\pm$0.02
DR-MAML (Two-Stage, Ours)	**0.89**$\pm$**0.04**	**2.91**$\pm$**0.46**	**1.76**$\pm$**0.02**	**0.54**$\pm$**0.01**	**1.70**$\pm$**0.17**	**0.96**$\pm$**0.01**

Table 5: **Average `5-way 1-shot` classification accuracies in *mini*-ImageNet with reported standard deviations (3 runs). We mainly compare DR-MAML with different optimization algorithms.** With $\alpha = 0.5$ for meta training, the best testing results are in bold.

Method	Eight Meta-Training Tasks			Four Meta-Testing Tasks		
	Average	Worst	CVaR_α	Average	Worst	CVaR_α
DR-MAML (Risk-Reweighted, (Sagawa et al., 2020))	67.0$\pm$0.2	56.6$\pm$0.4	61.6$\pm$0.2	49.1$\pm$0.2	46.6$\pm$0.1	47.2$\pm$0.2
DR-MAML (Two-Stage, Ours)	**70.2**$\pm$**0.2**	**63.4**$\pm$**0.2**	**67.2**$\pm$**0.1**	**49.4**$\pm$**0.1**	**47.1**$\pm$**0.1**	**47.5**$\pm$**0.1**

strategies consistently outperform that with the risk-weighted ones in both `5-shot` and `10-shot` sinusoid cases regarding all metrics. The performance advantage of using the two-stage ones is not significant in *mini*-ImageNet scenarios. We can hypothesize that the estimate of VaR_α in continuous task domains, *e.g.*, sinusoid regression, is more accurate, and this probabilistically ensures the improvement guarantee with two-stage strategies. Both the VaR_α estimate in two-stage strategies and the importance weight estimate in the risk-reweighted ones may have a lot of biases in few-shot image classification, which lead to comparable performance.

L Platforms & Computational Tools

In this research project, we use NVIDIA 1080-Ti GPUs in computation. Pytorch (Paszke et al., 2019) works as the deep learning toolkit in implementing few-shot image classification experiments. Meanwhile, Tensorflow is the deep learning toolkit for implementing sinusoid few-shot regression experiments.