Constrained Meta Agnostic Reinforcement Learning

Karam Daaboul[*]
KIT Karlsruhe Institute of Technology
Germany
`daaboul@kit.edu`

Florian Kuhm[*]
KIT Karlsruhe Institute of Technology
Germany
`florian.kuhm@student.kit.edu`

Tim Joseph
FZI Research Center for Information Technology
Germany
`joseph@fzi.de`

Marius J. Zöllner
KIT Karlsruhe Institute of Technology
Germany
`marius.zoellner@kit.edu`

Abstract: Meta-Reinforcement Learning (Meta-RL) aims to acquire meta-knowledge for quick adaptation to diverse tasks. However, applying these policies in real-world environments presents a significant challenge in balancing rapid adaptability with adherence to environmental constraints. Our novel approach, Constraint Model Agnostic Meta Learning (C-MAML), merges meta learning with constrained optimization to address this challenge. C-MAML enables rapid and efficient task adaptation by incorporating task-specific constraints directly into its meta-algorithm framework during the training phase. This fusion results in safer initial parameters for learning new tasks. We demonstrate the effectiveness of C-MAML in simulated locomotion tasks with mobile robots of varying complexity, where it outperforms standard Meta-RL methods.

Keywords: Safe Reinforcement Learning, Meta Reinforcement Learning

1 Introduction

Since its inception, Machine Learning, and particularly Reinforcement Learning (RL), have strived to emulate human learning processes in machines. A notable early example is Rosenblatt's development of the Perceptron in 1957, modeled after the human brain [1]. This approach continued to evolve, with RL drawing inspiration from the biological and psychological parallels observed in human and animal behavior [2]. Humans, unlike machines, begin learning new tasks not from scratch but by harnessing previously acquired knowledge, thus optimizing the learning process and ensuring safety. For instance, the knowledge of vehicle dynamics and road safety rules significantly accelerates learning how to drive a new car, showcasing a blend of adaptation and safety considerations. Similarly, safe adaptation is crucial in robotic systems to prevent harm to both the environment and the robot itself. Addressing this challenge, the field of meta learning has emerged, producing a diverse array of algorithms aimed at mimicking this human-like efficiency in learning [3, 4, 5]. The objective is to train RL agents to adapt rapidly to various tasks with minimal new task-specific experiences. This adaptability was illustrated by Beck et al. (2023) with the example of a cooking robot that, during Meta-Training, learns to navigate different kitchen layouts and appliance arrangements. Upon deployment in a customer's kitchen, the robot adjusts to this new environment relatively easily.

However, a critical issue largely overlooked in meta learning literature is the safety of the meta learning process itself. The agent should not behave in a manner that endangers itself or its surroundings, neither during the meta learning phase nor in subsequent deployment. For instance, a kitchen robot

---

[*]These authors contributed equally to this work.

must be able to recognize the presence of people to prevent injuring them, even while adapting to an unfamiliar kitchen layout. While there is substantial safety-oriented research in RL [7, 8, 9, 10], these studies typically focus on adhering to constraints during the training of an agent for a specific task. In contrast, this work presents a method that ensures safety during meta-training, generates a safe set of initial parameters and adheres to constraints during fine-tuning. The combination of safety and meta learning presents distinct challenges that have not been fully addressed in existing literature. The presented research introduces Constrained MAML (C-MAML), an innovative approach designed to improve safety during training and adaptation phases in meta-learning. This enhancement is achieved by incorporating constraint-based methods into the traditional Model Agnostic Meta Learning (MAML) framework. Additionally, we have expanded the C-MAML framework by developing a practical algorithm that incorporates first-order meta-gradient methods along with a global safety critic in the outer loop. This enhancement is designed to ensure computational efficiency and the safety of the learned meta-policy during its application. Furthermore, the results demonstrate the adaptability of C-MAML, highlighting its capability to function effectively with various safe RL methods in the inner loop. The paper's structure is as follows: Section 2 presents the state-of-the-art solutions in meta learning and safe RL, forming the foundation for our newly developed algorithm. Section 3 introduces the essential preliminaries and notation. Our novel approach, C-MAML, is detailed in Chapter 4, followed by experiments and an outlook in sections 5 and 6.

2 Related Work

2.1 Meta Learning

Meta-learning, or "learning to learn," aims at discovering a meta-initialization to enhance the speed of adaptation to new tasks, moving beyond traditional random initialization approaches [11]. In Meta-RL, this concept is extended to train a meta-policy across varied environments to facilitate quick adjustment to new challenges [12]. Early Meta Learning efforts, like those by Chalmers et al. (1991), utilized black-box strategies to evolve update rules for neural net weights. Meta-RL's black-box methods, such as RL^2 [14], embed the learning algorithm within an RNN, while gradient-based strategies like MAML prioritize swift adaptation via gradient updates. Finn et al.'s (2017) MAML optimizes a model's initial parameters for improved task performance through a dual-level optimization process, beneficial for both model-free [15] and model-based RL [16]. FoMAML, a variant by Finn et al. (2017), simplifies MAML by omitting second-order gradients, offering efficiency with minimal performance loss. Similarly, REPTILE by Nichol et al. (2018) updates meta-parameters without second-order derivatives. iMAML [5] addresses information loss by indirectly incorporating the Jacobi matrix in the meta-gradient calculation. While these methodologies excel in fast adaptability, they typically overlook safety considerations in learning initialization.

2.2 Safe Reinforcement Learning

Research in safe RL focuses on ensuring adherence to safety constraints through techniques like gradient or parameter projection, exemplified by Projection-Based Constrained Policy Optimization (PBCPO) [17] and related methods. Dalal et al. (2018) introduced a strategy employing safety layers to adjust actions per state, bypassing the need for domain-specific action modifications. Concurrently, Chow et al. (2018) explored using Linear Programming to derive Lyapunov functions for enforcing agent safety constraints. Lagrangian duality methods, addressing optimization while respecting constraints are pivotal. This category includes Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO), leveraging Lagrangian techniques [?]. Achiam et al. (2017) introduced Constrained Policy Optimization (CPO), a sophisticated second-order algorithm for identifying feasible policies within a trust region, ensuring both performance improvement and constraint adherence. Although these algorithms provide solutions for real-world RL applications, it is critical to have agents that can quickly adapt to new tasks in dynamic environments. This aspect is often overlooked in the pursuit of safety but is essential for practical deployment.

3   Preliminaries

Meta-RL aims to find a meta-policy that can efficiently adapt to a wide range of tasks from a task distribution T, aiming to maximize the expected return across these tasks. This is formalized as:

$$\max_\pi \ \mathbb{E}_{p \sim T} (J_p(\pi_p)). \tag{1}$$

Here, π denotes the meta-policy, which is adapted for each task p in T to a task-specific policy π_p, crucial for optimal performance across tasks. The adaptation of π to π_p for task p is achieved through an update mechanism U_p:

$$\pi_p = U_p(\pi). \tag{2}$$

In practice, U_p might utilize an algorithm like gradient descent [3], updating the meta-policy based on the gradient of $J_p(\pi)$:

$$\pi_p = \pi - \alpha \cdot \nabla_\pi J_p(\pi). \tag{3}$$

In the distribution T, tasks are modeled as unique Constrained Markov Decision Processes (CMDPs) represented by the tuple $(\mathcal{S}, \mathcal{A}, r_p, \mathcal{C}_p, M, \mu, D_p)$. Here, $\mathcal{S}$ and $\mathcal{A}$ signify the state and action spaces, respectively. The reward function specific to task p is $r_p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, while $\mathcal{C}_p$ consists of safety-related cost functions, with each $c_{p,j} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. The CMDP framework also includes the state transition probability $M : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ and the initial state distribution $\mu : \mathcal{S} \to [0, 1]$. Furthermore, D_p details the predefined cost limits $\{d_{p,j} \in \mathbb{R}\}$ for each cost function $c_{p,j}$ in task p. Although all tasks utilize the same M, they are distinguished by their specific reward and cost structures. The expected discounted returns and costs under a policy $\pi(a|s)$ for task p are given by:

$$J_p(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^\infty \gamma^t r_p(s_t, a_t, s_{t+1}) \right], \quad J_{c_{p,j}}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^\infty \gamma^t c_{p,j}(s_t, a_t, s_{t+1}) \right].$$

In this context, τ represents trajectories starting from initial states s_0, with actions a drawn according to $\pi(a|s)$, and subsequent states resulting from M. The discount factor is denoted by γ. A policy π is safe if it meets the condition $J_{c_{p,j}}(\pi) \le d_{p,j}$ for every j in $\mathcal{C}_p$, thus establishing a set of acceptable policies $\Pi_{\mathcal{C}_p}$ for task p. The aim of constrained policy optimization is thus to optimize expected returns within these safety limits:

$$\max_\pi J_p(\pi) \quad \text{s.t.} \quad J_{c_{p,j}}(\pi) \le d_{p,j}, \forall j \in \mathcal{C}_p.$$

Given $R_p(\tau) = \sum_{t=0}^\infty \gamma^t r_p(s_t, a_t, s_{t+1})$ as the return for a trajectory τ, the value function $V_p^\pi(s)$ and action-value function $Q_p^\pi(s, a)$ are defined as the expected returns from state s and from taking action a in state s under policy π, respectively:

$$V_p^\pi(s) = \mathbb{E}_{\tau \sim (\pi)} [R_p(\tau)|s_0 = s], \quad Q_p^\pi(s, a) = \mathbb{E}_{\tau \sim (\pi)} [R_p(\tau)|s_0 = s, a_0 = a].$$

The advantage function, $A_p^\pi(s, a) = Q_p^\pi(s, a) - V_p^\pi(s)$, quantifies the relative benefit of action a in state s under π. In addition to these functions, the CMDP setting also includes cost-related counterparts for each task p in T, defining the total trajectory cost as $C_{p,j}(\tau) = \sum_{t=0}^\infty \gamma^t \cdot c_{p,j}(s_t, a_t, s_{t+1})$. For expected cost assessments, it introduces $V_{c_{p,j}}^\pi(s)$, $Q_{c_{p,j}}^\pi(s, a)$, and $A_{c_{p,j}}^\pi(s, a)$ corresponding to each cost function $c_{p,j}$ in $\mathcal{C}_p$. For simplicity, subsequent discussions assume each task has only one constraint.

Analysis often employs the stationary discounted state distribution $\rho^\pi(s)$, the probability of visiting state s under policy π with transition probability P. It is defined as $\rho^\pi(s) = (1-\gamma) \sum_{t=0}^\infty \gamma^t P(s_t = s|\pi)$, where $P(s_t = s|\pi)$ is the probability of being in state s at time t following policy π.

Kakade and Langford (2002) describe the expected return difference between policies π and π' as:

$$J(\pi') = J(\pi) + \mathbb{E}_{\substack{s \sim \rho^{\pi'} \\ a \sim \pi'}} \left(\sum_{t=0}^\infty \gamma^t \cdot A^\pi(s, a) \right). \tag{4}$$
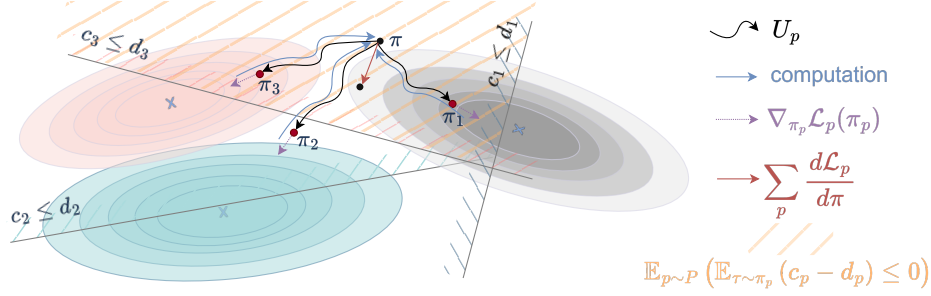
Figure 1: Visual representation of the Constrained Model Agnostic Meta Learning (C-MAML) framework. The meta-policy is trained across different tasks. Task-specific policies (π_1, π_2, π_3) are adjusted within their respective constraint surfaces C_1, C_2, C_3, each with a dedicated safety boundary d_1, d_2, d_3.

This captures expected returns' variation, combining state transitions and policy advantages. However, direct sampling for optimization is intractable. Schulman et al. (2015) used importance sampling to address this:

$$J(\pi') = J(\pi) + \mathop{\mathbb{E}}_{\substack{s \sim \rho^{\pi'} \\ a \sim \pi}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\pi'(a|s)}{\pi(a|s)} \cdot A^\pi(s, a) \right). \tag{5}$$

Additionally, they integrated this approach into a trust region optimization framework to facilitate sampling from the state distribution $\rho^{\pi'}$.

4   Constrained Model Agnostic Meta Reinforcement Learning

Our framework, Constrained Model Agnostic Meta Learning (C-MAML), introduces a gradient-based meta-learning approach designed to address both adaptability and safety in task learning. C-MAML incorporates task-specific constraints in the inner loop a universal safety constraint in the outer loop that integrates all individual task constraints across the task distribution. This method ensures that each task's learning process is sensitive to its specific context and demands, embedding safety at the core of the adaptation process.

4.1   Incorporating Constraints in the Inner Loop

Our framework models each task as a constrained Markov decision process (CMDP), aiming to optimize policies for maximum reward under specific constraints. The task-specific policy, denoted as π_p, is obtained by solving the following optimization problem:

$$\pi_p = \arg\max_{\tilde{\pi}} \mathop{\mathbb{E}}_{\tau \sim \tilde{\pi}} (R_p(\tau|s_0 = s)) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\tau \sim \tilde{\pi}} (C_p(\tau|s_0 = s)) - d_p \leq 0 \tag{6}$$

One effective method for solving this CMDP is through Trust Region Policy Optimization with Lagrangian methods (TRPOLag) [9]. TRPOLag extends the robust performance guarantees of TRPO by integrating a Lagrangian framework that specifically addresses constraints. This integration enables the reformulation of the objective in Equation (6) to optimize task-specific goals within defined safety constraints. The revised formulation is presented as follows:

$$\mathcal{L}_{\text{inner}} = \mathcal{L}_p(\pi_p, \lambda_p) = \mathop{\mathbb{E}}_{\tau \sim \pi} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\pi_p(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_p^\pi(s_t, a_t) \right) \tag{7}$$

$$- \lambda_p \cdot \left(\mathop{\mathbb{E}}_{\tau \sim \pi} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\pi_p(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_{C_p}^\pi(s_t, a_t) \right) + J_C(\pi) - d \right)$$

$$\text{s.t.} \quad D_{KL}(\pi_p \| \pi) \leq \epsilon$$

This formulation utilizes dual variables, λ_p, for each task p. The parameter ϵ serves as a predefined threshold for the KL divergence. For a detailed derivation of this equation, please refer to C.

4

4.2 Optimizing Meta-Parameter in the Outer Loop

Contrary to conventional Meta-RL algorithms that prioritize learning efficiency, our approach, C-MAML, integrates an intrinsic safety mechanism within the learning framework. This is accomplished by identifying a meta-parameter in the solution space that adheres to all constraints. This ensures that the selected initialization not only accelerates learning for newly encountered tasks in the inner loop but also complies with all constraints faced during the meta-training phase. This synthesis is encapsulated in the meta-objective function:

$$\max_{\pi} \ \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(R_p(\tau | s_0 = s) \right) \right) \quad \text{s.t.} \quad \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(C_p(\tau | s_0 = s) \right) - d_p \right) \le 0 \qquad (8)$$

4.3 Enhancing Safety of the Meta-Policy

Employing MAML to maximize the objective function (8) in the outer loop, introduces significant computational challenges due to its reliance on second-order gradients. This reliance arises because MAML requires computing the gradient of the loss with respect to the meta-parameters after a gradient update on the task-specific loss. This complexity is magnified when integrating inner loop constraints, making the outer loop's meta-policy π optimization notably intensive. A practical solution is to utilize First-Order Model-Agnostic Meta-Learning (FoMAML), which simplifies the gradient estimation process by avoiding second-order derivatives. This approach enables more manageable updates to π based on outcomes from task-specific policies. However, it is crucial to incorporate safety information derived during the inner loop's computations. Without this integration, the meta-policy might not consistently maintain safety standards across different task-specific policies despite its flexibility.

$$\max_{\pi} \ \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(R_p(\tau) \right) \right) \quad \text{s.t.} \quad \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(C_p(\tau | s_0 = s) \right) - d_p \right) \le 0 \qquad (9)$$

$$\mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi} \left(C_p(\tau | s_0 = s) \right) - d_p \right) \le 0.$$

This secondary constraint in (9) ensures that π consistently exhibits safe behavior across all tasks, leading to a modified Lagrangian optimization for the outer loop:

$$\mathcal{L}_{\text{outer}}(\pi, \lambda, \eta) = \mathbb{E}_{p \sim T} \left[\mathbb{E}_{\tau \sim \pi_p} \left(R_p(\tau | s_0 = s) \right) - \lambda \left(\mathbb{E}_{\tau \sim \pi_p} \left(C_p(\tau | s_0 = s) \right) - d_p \right) \right. \qquad (10)$$

$$\left. - \eta \left(\mathbb{E}_{\tau \sim \pi} \left(C_p(\tau | s_0 = s) \right) - d_p \right) \right].$$

Incorporating η as a crucial safety regulator enables the meta-policy π to efficiently adapt to new tasks while steadfastly upholding safe practices across different task environments. Section **??** empirically validates these benefits, demonstrating the significant advantage of integrating η into the outer loop optimization. Ensuring safety across the diverse range of tasks requires finding an initial setting (meta-parameter) that satisfies the unique constraints of each task. However, conflicting constraints among tasks make it challenging to identify a universally safe meta-parameter across the parameter space. To overcome this challenge and enable the discovery of such a meta-parameter, we refine the objective function (10). Initially configured for task-specific constraints c_p and their respective thresholds d_p, our approach focuses on adopting constraints c and thresholds d independent of any specific task. Such a substitution is possible when safety requirements are common across tasks, allowing a unified constraint to represent multiple task-specific constraints. For example, suppose all tasks involve robotic manipulation where joint limits must not be exceeded to prevent damage. In that case, a general constraint on joint angles can replace individual constraints for each task. However, this substitution must ensure that the general constraint is equally or more restrictive than the most stringent task-specific constraint to ensure safety across all tasks. Should the general constraint be less restrictive, it risks inadequately enforcing safety requirements for some tasks.
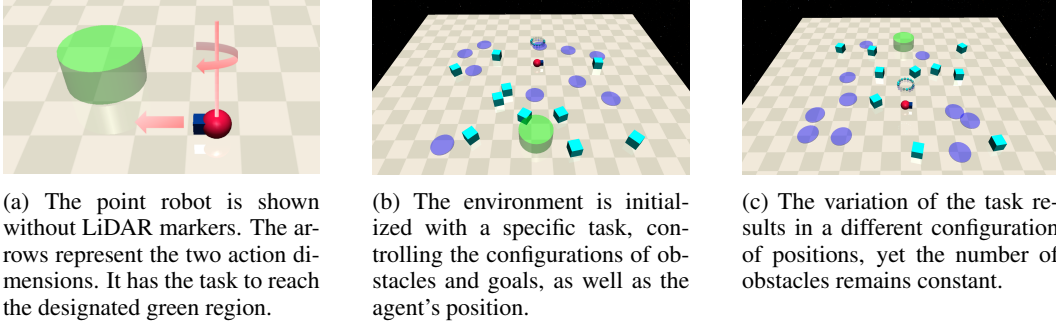
(a) The point robot is shown without LiDAR markers. The arrows represent the two action dimensions. It has the task to reach the designated green region.

(b) The environment is initialized with a specific task, controlling the configurations of obstacles and goals, as well as the agent's position.

(c) The variation of the task results in a different configuration of positions, yet the number of obstacles remains constant.

Figure 2: Illustrations of the action space and two different tasks of the used environment.

$$\mathcal{L}_{\text{outer}}(\pi, \lambda, \eta) = \mathop{\mathbb{E}}_{p \sim T} \left[\mathop{\mathbb{E}}_{\tau \sim \pi_p} (R_p(\tau|s_0 = s)) - \lambda \left(\mathop{\mathbb{E}}_{\tau \sim \pi_p} (C(\tau|s_0 = s)) - d \right) \right. \tag{11}$$
$$\left. - \eta \left(\mathop{\mathbb{E}}_{\tau \sim \pi} (C(\tau|s_0 = s)) - d \right) \right].$$

Leveraging Assumption (11) enables the development of a meta cost critic, which can be trained with on-policy data gathered by the meta-policy across a variety of encountered tasks.

$$V_C^\pi(s) = \mathop{\mathbb{E}}_{\tau \sim \pi} (C(\tau|s_0 = s)). \tag{12}$$

Integrating the meta cost critic, the updated objective function is introduced as follows:

$$\mathcal{L}_{\text{outer}}(\pi, \lambda, \eta) = \mathop{\mathbb{E}}_{p \sim T} \left[\mathop{\mathbb{E}}_{\tau \sim \pi_p} (R_p(\tau|s_0 = s)) - \lambda \left(\mathop{\mathbb{E}}_{\tau \sim \pi_p} (C(\tau|s_0 = s)) - d \right) - \eta \left(V_C^\pi(s) - d \right) \right]. \tag{13}$$

Calculating the gradient of the objective function (13) with respect to the meta-policy π returns:

$$\nabla_\pi \mathcal{L}_{\text{outer}} \approx I \cdot \mathop{\mathbb{E}}_{p \sim T} (\nabla_\pi \mathcal{L}_{\text{outer}}(\pi_p)) - \frac{\partial}{\partial \pi} \eta \underbrace{\mathop{\mathbb{E}}_{\tau \sim \pi} \left(\sum_{t=0}^\infty \log(\pi(a_t|s_t)) \cdot V_c^\pi(s) \right)}_{\text{Independent of } \pi_p} \tag{14}$$

Here, the identity matrix I approximates the Hessian, simplifying the gradient computation process by assuming constant second derivatives and treating the loss surface as locally linear. This substitution significantly reduces computational complexity, making the optimization more tractable for high-dimensional problems. This approximation works reasonably well when the loss surface is relatively smooth, and the second-order terms do not vary significantly. The gradient $\nabla_\pi \mathcal{L}_{\text{outer}}(\pi_p)$ corresponds to FoMAML's return gradient, which balances rewards and costs. The second term, independent of task-specific updates π_p, significantly affects the meta-policy's adaptation by promoting safer actions as evaluated by the safety critic. This effectively increases the propensity for actions to reduce costs and diminishes those associated with higher costs. A detailed gradient derivation is provided in Appendix D, with a complete method diagram and pseudocode in Appendix E.

5 Evaluation

We evaluated the efficacy of C-MAML through simulations involving mobile tasks in the *Safety-Gymnasium* environment [21], which provides pre-implemented costs and rewards. The benchmark involves navigating a spherical robot towards a goal while avoiding obstacles like hazards (static) and vases (movable cubes) that penalize the agent upon contact or movement. The robot moves in the x-y plane by adjusting its velocity and rotating around the z-axis (see Figure 2). Its two-dimensional action space and receives rewards based on proximity to the goal, calculated as the distance difference before and after an action. The key questions addressed include:
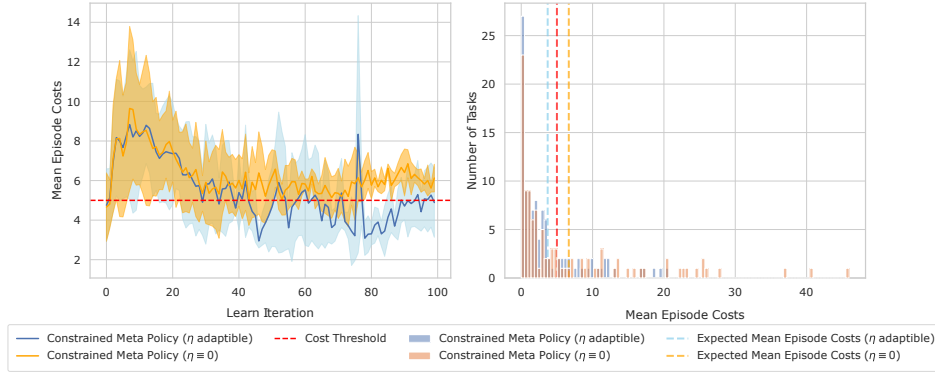
Figure 3: Evaluation of η on policy safety and adaptability: On the left, meta-training performance across 106 tasks, showing the effect of an adaptive η (employing safety critic) versus $\eta = 0$ (no safety critic) on maintaining safer cost margins. On the right, fine-tuning phase performance, illustrating how an adaptive η contributes to consistent adherence to the $d = 5$ cost threshold compared to the absence of a safety critic.

- How does C-MAML ensure safety during **training** using first-order meta-learning and a safety critic, and how does this impact policy safety and adaptability during **fine-tuning**?
- What effects do safety constraints integrated into meta-initialization parameters have on adaptation speed and cost control in C-MAML compared to traditional methods?
- Is C-MAML **agnostic** to the specific safe RL methods utilized in the inner loop?

Utilizing First-Order Model Agnostic Meta Learning (FoMAML) instead of conventional MAML leads to the omission of valuable second-order information during the inner loop's optimization phase. To mitigate this, we introduce a new constraint, regulated by the Lagrange multiplier η, into the optimization objective (13). Our evaluation centers on assessing the impact of η on policy safety and adaptability during both meta-training and fine-tuning stages. This involves a comparative analysis between two methodologies: one employing a trainable η and the other proceeding without it (Figure 3). During meta-training, the adaptive η, guided by the safety critic, helped maintain safer cost margins, avoiding the volatility observed with a fixed η at zero. In the fine-tuning phase, although both approaches occasionally breached the $d = 5$ cost limit, employing an adaptive η resulted in more consistent adherence to the cost threshold. This underscores the value of integrating an adaptive η and safety critic, ensuring safer policy adaptation and fine-tuning.

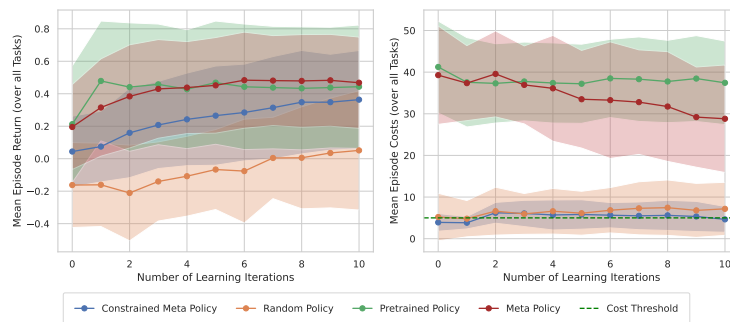## 5.1 Safety Adaptation Across Task Spectrum

Figure 4: Mean episode return and costs during fine-tuning across tasks. Policies are color-coded as follows: C-MAML with TRPOLag in the inner loop is depicted in blue, the randomly initialized policy in orange, the TRPOLag pretrained policy in green, and the MAML policy with TRPO in the inner loop is shown in red, highlighting the diverse adaptation strategies explored.

7

In the fine-tuning phase, the Trust Region Policy Optimization-Lagrangian (TRPOLag) algorithm was utilized, focusing on four policy initializations: randomly initialized, TRPOLag pretrained policy, MAML with TRPO in the inner loop, and C-MAML with TRPOLag in the inner loop. Our analysis across 106 tasks in Environment 2, shown in Figure 4, includes episode returns and costs with confidence bands indicating standard deviation. Compared to random initialization, the C-MAML initialization achieved faster adaptation, maintaining episode costs close to the cost limit ($d = 5$). Conversely, the TRPOLag pretrained policy and the unconstrained MAML initialization adapted more quickly but incurred higher costs during fine-tuning, highlighting their lack of safety considerations. This emphasizes the significance of incorporating safety into the adaptation process, as exemplified by C-MAML, to ensure both efficient and responsible policy development. Additional TRPOLag findings are detailed in Appendices H, G and J.
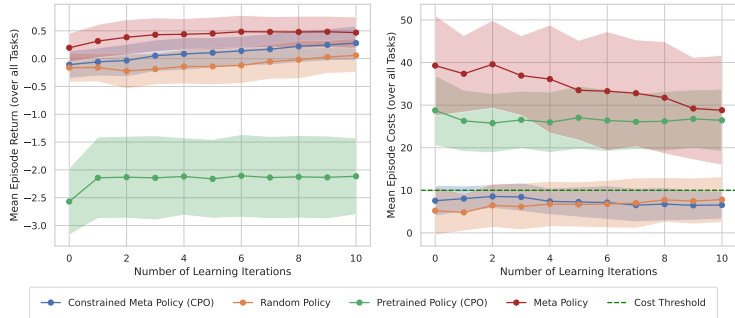
5.2 Model Agnosticity in C-MAML Framework

Figure 5: Mean episode return and costs during fine-tuning across tasks: C-MAML with CPO in the inner loop is depicted in blue, the randomly initialized policy in orange, CPO pretrained in green, and the MAML policy with TRPO in the inner loop is shown in red. Each of these policies (initialization) is fine-tuned using CPO.

Adhering to the MAML concept of model agnosticism, our C-MAML framework is crafted to function independently from any particular algorithm within its inner optimization loop, facilitating the use of various constrained policy optimization strategies. To illustrate, we implemented Constrained Policy Optimization (CPO) as the inner loop algorithm. Results depicted in Figure 5 reveal a performance trajectory consistent with those seen using C-MAML TRPOLag (Figure 4), underscoring our framework's versatility and efficacy with different constrained policy optimization approaches. C-MAML CPO adheres to the $d = 10$ cost limit, demonstrating efficient adaptability across tasks. For expanded insights on employing CPO in the inner loop, see Appendix I.

6 Conclusion

We presented Constrained Model-Agnostic Meta-Learning (C-MAML), a novel meta-RL framework that integrates gradient-based meta-learning with safe reinforcement learning to prioritize policy safety. It incorporates safe RL within its inner loop, resulting in policies that are adaptable and comply with safety norms. Our method enhances meta-policy safety by introducing additional constraints in the outer loop and employing a safety critic, ensuring adaptations prioritize safety. Our experiments demonstrate that C-MAML leads to safer and more adaptable policies compared to random or pre-trained starts. Future research could investigate other first-order meta-learning techniques and safe RL algorithms in the inner loop.

References

[1] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.

[2] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[3] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

[4] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[5] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *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 113–124, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/072b030ba126b2f4b2374f342be9ed44-Abstract.html.

[6] J. Beck, R. Vuorio, E. Z. Liu, Z. Xiong, L. M. Zintgraf, C. Finn, and S. Whiteson. A survey of meta-reinforcement learning. *CoRR*, abs/2301.08028, 2023. doi:10.48550/ARXIV.2301.08028.

[7] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

[8] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 2017.

[9] A. Ray, J. Achiam, and D. Amodei. Benchmarking Safe Exploration in Deep Reinforcement Learning. *OpenAI*, 2019.

[10] M. A. Zanger, K. Daaboul, and J. Marius Zöllner. Safe continuous control with constrained model-based policy optimization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3512–3519, 2021. doi:10.1109/IROS51168.2021.9635984.

[11] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5149–5169, 2022. doi:10.1109/TPAMI.2021.3079209. URL https://doi.org/10.1109/TPAMI.2021.3079209.

[12] Y. Lee and S. Choi. Gradient-based meta-learning with learned layerwise metric and subspace. *35th International Conference on Machine Learning, ICML 2018*, 7:4574–4586, 2018.

[13] D. J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Connectionist Models*, pages 81–90. Morgan Kaufmann, 1991. ISBN 978-1-4832-1448-1.

[14] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016.

[15] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine. Meta-reinforcement learning of structured exploration strategies. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/4de754248c196c85ee4fbdcee89179bd-Paper.pdf.

[16] K. Daaboul, J. Ikels, and J. M. Zöllner. Robotic control using model based meta adaption. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9457–9463, 2023. doi:10.1109/ICRA48891.2023.10160425.

[17] T. Yang, J. Rosca, K. Narasimhan, and P. Ramadge. Projection-based constrained policy optimization. 2020. Funding Information: The authors would like to thank the anonymous reviewers and the area chair for their comments. Tsung-Yen Yang thanks Siemens Corporation, Corporate Technology for their support. Publisher Copyright: © 2020 8th International Conference on Learning Representations, ICLR 2020. All rights reserved.; 8th International Conference on Learning Representations, ICLR 2020 ; Conference date: 30-04-2020.

[18] G. Dalal, K. Dvijotham, M. Vecerík, T. Hester, C. Paduraru, and Y. Tassa. Safe exploration in continuous action spaces. *CoRR*, abs/1801.08757, 2018.

[19] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[20] S. M. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In C. Sammut and A. G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002.

[21] J. Ji, B. Zhang, J. Zhou, X. Pan, W. Huang, R. Sun, Y. Geng, Y. Zhong, J. Dai, and Y. Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

A   Notation

To ensure clarity and precision in the equations throughout this document, we adopt the following notational conventions:

- $\tau \sim \pi$ indicates that the trajectory τ is sampled according to the policy π, where actions a are sampled from π and states s follow the discounted state distribution ρ^π.
- When actions a and states s are sampled from different policies or distributions, we explicitly specify each source in the equations. For example, $a \sim \pi$ and $s \sim \rho^{\tilde{\pi}}$ indicates that actions are sampled according to policy π while states follow the distribution associated with a different policy $\tilde{\pi}$.

B   Mathematical Formulation of Constrained Meta Reinforcement Learning Objective

In the domain of constrained Meta-RL, the overarching objective is to identify a meta-policy that is not only swiftly adaptable to new tasks but also maintains safety. The formal mathematical representation of this objective, along with the associated constraints, is detailed below.

Primary Objective and Constraints: The primary goal in our approach is to maximize the expected cumulative discounted reward while ensuring that the accumulated discounted costs across tasks do not exceed a specific threshold. This is formalized in the following mathematical structure:

$$\max_{\pi} \quad \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\substack{s_0 \sim \mu_p \\ a_t \sim \pi_p(\cdot|s_t) \\ s_{t+1} \sim M(\cdot|s_t,a_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot r_p(s_t, a_t, s_{t+1}) \right) \right) \tag{15}$$

$$s.t. \quad \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\substack{s_0 \sim \mu_p \\ a_t \sim \pi_p(\cdot|s_t) \\ s_{t+1} \sim M(\cdot|s_t,a_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot c_p(s_t, a_t, s_{t+1}) \right) - d_p \right) \leq 0.$$

For clarity, $R_p(\tau)$ represents the expected value of discounted trajectory returns for task p, and $C_p(\tau)$ indicates the expected value of discounted trajectory costs for the same task. The trajectories τ are sampled from the policy π_p, which is a task-specific policy derived from the general meta-policy π.

The reformulated objective is then expressed as:

$$\max_{\pi} \quad \mathbb{E}_{p \sim T} (\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau))) \tag{16}$$
$$s.t. \quad \mathbb{E}_{p \sim T} (\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau))) \leq d_p$$

In this work, we consider a special case of the optimization problem mentioned above. Although problem 16 accommodates task-specific cost functions c_p and thresholds d_p, in practice, constraints that are uniform across all tasks are often more applicable. For instance, in robotic applications, there may be universal constraints such as preventing self-damage due to excessive joint twisting. Therefore, in the subsequent analysis, we explore a scenario with task-independent constraints, simplifying the model to accommodate a single, overarching constraint that applies uniformly to all tasks.

$$\max_{\pi} \quad \mathbb{E}_{p \sim T} (\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau))) \tag{17}$$
$$s.t. \quad \mathbb{E}_{p \sim T} (\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau))) \leq d$$

Lagrangian Formulation in the Inner Loop: In our approach, the Lagrangian method plays a pivotal role in the inner loop optimization. Theoretically, if we consider θ'_p as the actual stationary

point, we encounter the following inner loop optimization problem:

$$\pi_p = \arg\min_{\lambda \geq 0} \max_{\tilde{\pi}} \quad \mathbb{E}_{\tau \sim \tilde{\pi}} (R_p(\tau)) - \lambda \mathbb{E}_{\tau \sim \tilde{\pi}} (C_p(\tau) - d) \tag{18}$$

Since $\tilde{\pi}$ is an unknown policy to be determined, we can utilize the inequality by Kakade and Langford (2002) to consider a surrogate function, akin to the one used in TRPO [7], which approximately corresponds to the above objective function. This eliminates the need to generate rollouts from a policy $\tilde{\pi}$ to estimate the expected value.

$$
\begin{aligned}
\pi_p = \arg\min_{\lambda \geq 0} \max_{\tilde{\pi}} \quad & \mathbb{E}_{\substack{s_t \sim \rho^{\tilde{\pi}} \\ a_t \sim \tilde{\pi}(\cdot|s_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \cdot A^{\pi}(s_t, a_t) \right] \\
& - \lambda \left(\mathbb{E}_{\substack{s_t \sim \rho^{\tilde{\pi}} \\ a_t \sim \tilde{\pi}(\cdot|s_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \cdot A_C^{\pi}(s_t, a_t) \right] - d \right) \\
\overset{\substack{\text{Importance} \\ \text{Sampling}}}{=} \arg\min_{\lambda \geq 0} \max_{\tilde{\pi}} \quad & \mathbb{E}_{\substack{s_t \sim \rho^{\tilde{\pi}} \\ a_t \sim \pi(\cdot|s_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A^{\pi}(s_t, a_t) \right) \\
& - \lambda \left(\mathbb{E}_{\substack{s_t \sim \rho^{\tilde{\pi}} \\ a_t \sim \pi(\cdot|s_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_C^{\pi}(s_t, a_t) \right) - d \right) \\
\overset{\text{TRPO}}{\approx} \arg\min_{\lambda \geq 0} \max_{\pi} \quad & \mathbb{E}_{\substack{s_t \sim \rho^{\pi} \\ a_t \sim \pi(\cdot|s_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A^{\pi}(s_t, a_t) \right) \\
& - \lambda \left(\mathbb{E}_{\substack{s_t \sim \rho^{\pi} \\ a_t \sim \pi(\cdot|s_t)}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_C^{\pi}(s_t, a_t) \right) - d \right)
\end{aligned}
\tag{19}
$$

$$s.t. \quad D_{KL}(\tilde{\pi}, \pi) \leq \epsilon$$

By employing a primal-dual gradient approach, the optimization can be iteratively performed for both $\tilde{\pi}$ and λ. This optimization problem must be resolved in the inner loop of the MAML-TRPOLag algorithm for each selected task p.

First Order Optimization in Outer Loop: Transforming equation (17) from constrained to unconstrained optimization using the Lagrangian method, we derive the following outer loop optimization problem:

$$\mathcal{L}_{\text{outer}} = \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau)) - \lambda \left(\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right) \right) \tag{20}$$

This equation reveals the presence of two meta-parameters: the policy parameter π and the Lagrange multiplier λ. Therefore, it is necessary to compute the partial derivative for each of these parameters.

The calculation of the meta-gradient with respect to the meta-policy π is as follows:

$$
\begin{aligned}
\nabla_{\pi} \mathcal{L}_{\text{outer}} &= \frac{\partial}{\partial \pi} \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau)) - \lambda \left(\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right) \right). \\
&= \frac{\partial \pi_p}{\partial \pi} \frac{\partial}{\partial \pi_p} \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau)) - \lambda \left(\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right) \right) \\
&= \frac{\partial U_p(\pi)}{\partial \pi} \cdot \mathbb{E}_{p \sim P} \left(\frac{\partial}{\partial \pi_p} \mathbb{E}_{\tau \sim \pi_p} (R_p(\tau)) - \lambda \cdot \frac{\partial}{\partial \pi_p} \mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right) \\
&\overset{\substack{\text{Policy} \\ \text{Gradient}}}{=} \frac{\partial U_p(\pi)}{\partial \pi} \cdot \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot R_p(\tau) \right) - \lambda \cdot \mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot C_p(\tau) \right) \right)
\end{aligned}
$$

In the First order MAML (FoMAML) framework, the derivative of the update rule $U_p(\pi)$ is approximated as an identity matrix, simplifying the gradient computation. This approximation is essential

for efficiently updating the meta-policy, aligning with the objective and constraints.

$$
\begin{aligned}
\nabla_\pi \mathcal{L}_{\text{outer}} &= \frac{\partial U_p(\pi)}{\partial \pi} \cdot \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot R_p(\tau) \right) - \lambda \cdot \mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot C_p(\tau) \right) \right) \\
&\overset{\text{FoMAML}}{\approx} I \cdot \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot R_p(\tau) \right) - \lambda \cdot \mathbb{E}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot C_p(\tau) \right) \right)
\end{aligned}
\tag{21}
$$

For the meta Lagrange parameter λ, the gradient is calculated as follows:

$$
\begin{aligned}
\nabla_\lambda \mathcal{L}_{\text{outer}} &= \frac{\partial}{\partial \lambda} \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} (R_p(\tau)) - \lambda \left(\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right) \right) \\
&= - \mathbb{E}_{p \sim T} \left(\mathbb{E}_{\tau \sim \pi_p} (C_p(\tau) - d) \right),
\end{aligned}
\tag{22}
$$

such that the constraint $\lambda \geq 0$ is maintained.

C   Solving CMDP with TRPO and Lagrangian Methods

In the domain of Constrained Markov Decision Processes (CMDP), we aim to optimize policies that maximize expected rewards while adhering to predefined constraints on expected costs. The typical CMDP problem is formulated as follows:

$$
\pi^* = \arg\max_{\tilde{\pi}} \mathbb{E}_{\tau \sim \tilde{\pi}} [R(\tau|s_0 = s)] \quad \text{s.t.} \quad \mathbb{E}_{\tau \sim \tilde{\pi}} [C(\tau|s_0 = s)] \leq d,
\tag{23}
$$

where π^* denotes the optimal policy, $R(\tau|s_0 = s)$ and $C(\tau|s_0 = s)$ represent the total reward and cost from trajectory τ, respectively, and d is the allowable cost threshold.

To transform this CMDP into an unconstrained optimization problem, we employ the Lagrangian method by introducing a Lagrange multiplier λ:

$$
\mathcal{L}(\pi, \lambda) = \mathbb{E}_{\tau \sim \pi} [R(\tau|s_0 = s)] - \lambda \left(\mathbb{E}_{\tau \sim \pi} [C(\tau|s_0 = s)] - d \right),
\tag{24}
$$

where λ penalizes the policy when the expected cost exceeds the threshold d.

C.1   Trust Region Policy Optimization (TRPO)

Trust Region Policy Optimization (TRPO) addresses the optimization of policies in Markov Decision Processes (MDPs) by introducing a KL divergence constraint. This constraint ensures that policy updates remain within a predefined "trust region", thereby promoting stability. The objective for unconstrained policy improvement using TRPO is given by:

$$
\pi^* = \arg\max_{\tilde{\pi}} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A^\pi(s, a) \right] \quad \text{s.t.} \quad D_{KL}(\pi \| \tilde{\pi}) \leq \epsilon,
\tag{25}
$$

where $A^\pi(s, a)$ is the advantage function of the old policy π, and δ denotes the size of the trust region as determined by the KL divergence threshold.

C.2   Incorporating Cost in Policy Optimization Strategies

Building upon TRPO, [8] introduced an extended framework that accommodates policy constraints. This approach equates the policy's performance regarding rewards to a similar measure for the expected value of accumulated future costs:

$$
J_C(\pi) = \mathbb{E}_{\tau \sim \pi} (C(\tau) - d).
$$

The performance concerning costs can also be approximated as:

$$J_C(\tilde{\pi}) = J_C(\pi) + \underset{\substack{a \sim \pi \\ s \sim \rho^{\tilde{\pi}}}}{\mathbb{E}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_C^\pi(s_t, a_t) \right),$$

where $A_c^\pi(s, a)$ is the advantage cost function of the old policy π.

Assuming the same approximation of discounted state distribution $\rho^{\tilde{\pi}}$ as ρ^π given the KL divergence constraint between policies, the resulting optimization problem becomes:

$$\begin{aligned}
\max_{\tilde{\pi}} \quad & E_{\substack{a \sim \pi \\ s \sim \rho^\pi}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A^\pi(s_t, a_t) \right) \quad (26) \\
\text{s.t.} \quad & J_C(\pi) + E_{\substack{a \sim \pi \\ s \sim \rho^\pi}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_C^\pi(s_t, a_t) \right) \leq d \\
& D_{KL}(\tilde{\pi} \| \pi) \leq \delta.
\end{aligned}$$

C.3 Integrating TRPO with Lagrangian (TRPOLag)

A further extension is discussed in Ray et al. (2019) [9], which employs the Lagrangian method to simplify CMDP solutions. This adaptation modifies the TRPO objective to incorporate both reward maximization and constraint adherence by adjusting the Lagrange multiplier λ:

$$\begin{aligned}
\mathcal{L}(\pi, \lambda) = \quad & \underset{\tau \sim \pi}{\mathbb{E}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_p^\pi(s_t, a_t) \right) \quad (27) \\
& - \lambda_p \cdot \left(\underset{\tau \sim \pi}{\mathbb{E}} \left(\sum_{t=0}^{\infty} \gamma^t \cdot \frac{\tilde{\pi}(a_t|s_t)}{\pi(a_t|s_t)} \cdot A_{c_p}^\pi(s_t, a_t) \right) + J_C(\pi) - d \right) \\
\text{s.t.} \quad & D_{KL}(\tilde{\pi} \| \pi) \leq \epsilon
\end{aligned}$$

where $A_R^{\pi_{\text{old}}}(s, a)$ and $A_C^{\pi_{\text{old}}}(s, a)$ are the advantage functions for the reward and cost, respectively.

This approach allows TRPOLag to effectively balance the trade-offs between maximizing rewards and adhering to cost constraints, providing a robust solution framework for CMDPs.

D Mathematical Formulation of the Practical Algorithm

Our practical algorithm adapts the objective function (17) of the method by adding a new constraint to ensure that the new meta-policy adheres to universal constraints.

$$\begin{aligned}
\max_{\pi} \quad & \underset{p \sim T}{\mathbb{E}} (\underset{\tau \sim \pi_p}{\mathbb{E}} (R_p(\tau))) \quad (28) \\
\text{s.t.} \quad & \underset{p \sim T}{\mathbb{E}} (\underset{\tau \sim \pi_p}{\mathbb{E}} (C_p(\tau)) \leq d) \\
& \underset{p \sim T}{\mathbb{E}} (\underset{\tau \sim \pi}{\mathbb{E}} (C_p(\tau)) \leq d)
\end{aligned}$$

By rewriting Equation (28) using Lagrangian formulation, we obtain an unconstrained objective. Additionally, by using the meta safety critic as an approximation for the expected cumulative sum of the meta-policy, we arrive at the following equation.

$$\mathcal{L}_{\text{outer}} = \underset{p \sim T}{\mathbb{E}} \left(\underset{\tau \sim \pi_p}{\mathbb{E}} (R_p(\tau)) \right) - \lambda \underset{p \sim T}{\mathbb{E}} \left(\underset{\tau \sim \pi_p}{\mathbb{E}} (C_p(\tau)) - d \right) - \eta \underset{\tau \sim \pi}{\mathbb{E}} (V_c^\pi(s)) - d \quad (29)$$

The gradient of the objective function with respect to the meta-policy π is approximated as follows:

$$\nabla_\pi \mathcal{L}_{\text{outer}} \approx I \cdot \mathop{\mathbb{E}}_{p \sim T} \left(\underbrace{\mathop{\mathbb{E}}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot R_p(\tau) \right) - \lambda \cdot \mathop{\mathbb{E}}_{\tau \sim \pi_p} \left(\sum_{t=0}^{\infty} \nabla_{\pi_p} \log(\pi_p(a_t|s_t)) \cdot C_p(\tau) \right)}_{\text{FoMAML's return gradient balancing rewards and costs}} \right)$$

$$- \frac{\partial}{\partial \pi} \eta \underbrace{\mathop{\mathbb{E}}_{\tau \sim \pi} \left(\sum_{t=0}^{\infty} \log(\pi(a_t|s_t)) \cdot V_c^\pi(s) \right)}_{\text{Independent of } \pi_p} \tag{30}$$

In this formulation, the first term under the approximation symbol represents the gradient derived from FoMAML, which balances rewards and costs. The second term emphasizes the contribution of the meta safety critic $V_c^\pi(s)$ to the overall gradient with respect to π.

For the meta Lagrange parameter λ, the gradient is calculated as in the equation (22). An additional gradient is computed to optimize η:

$$\nabla_\eta \mathcal{L}_{\text{outer}} = \mathop{\mathbb{E}}_{\tau \sim \pi} \left(V_c^\pi(s) \right) - d \tag{31}$$

E   C-MAML with η and Safety Critic

The practical algorithm utilizes a first-order meta-learning approach, integrating safety-critic with the Lagrangian multiplier η for enhanced safety. This is illustrated in Figure 6.
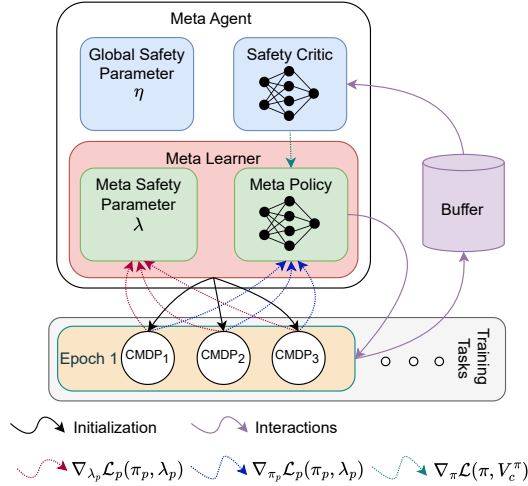

Figure 6: Diagram depicting the TRPOLag-based algorithm's execution: Initialization involves setting up task-specific policies π_p and safety parameters λ_p for each CMPD$_p$, derived from the meta-policy π and meta-parameter λ (**black arrows**). Data collection for each task employs these initial parameters (**purple arrows**). This data informs the refinement of the meta safety critic and the task-specific policy updates π_p. Feedback from task performances, encapsulating both reward and safety, guides the meta-policy π updates (**blue dotted arrows**). Simultaneously, the meta-policy is reinforced by the meta safety critic to align with overarching constraints (**green dotted arrows**), while aggregate signals from λ_p adjustments inform the meta-parameter λ updates (**red dotted arrows**).

The pseudocode detailing our method is presented below.

15

Algorithm 1 Constrained Model Agnostic Meta Learning (C-MAML)

1: **Input:** Number of outer iteration steps N, number of tasks for computing the meta-gradient B
2: **Output:** Meta-Policy π_θ and Cost Critic $V_{C,\hat{\theta}}$
3: Initialize the meta-policy π_θ and the meta-cost critic $V_{C,\hat{\theta}}$ randomly
4: **for** counter $\leq N$ **do**
5:     Randomly select a subset $\mathcal{P}$ of the support of P with $|\mathcal{P}| = B$
6:     **for** task p in $\mathcal{P}$ **do**
7:         $\pi'_p := \pi_\theta, V_{C,p} := V_{C,\hat{\theta}}$
8:         Randomly initialize the Reward Critic $V_{R,p}$
9:         **for** number of adaptation steps **do**
10:             Generate rollouts with π'_p
11:             Update the policy π'_p using a constrained optimization algorithm
12:             Update the critics $V_{C,p}$ and $V_{R,p}$
13:         **end for**
14:         Compute the meta-gradient for task p for the meta-policy and the meta-cost critic
15:     **end for**
16:     Average the task-specific meta-gradients and update π_θ and $V_{C,\hat{\theta}}$ using a meta-learning algorithm
17: **end for**
18: **return** $\pi_\theta, V_{C,\hat{\theta}}$

F   Task Distributions

The positions of the agent, obstacles, and the goal vary randomly with each task. The seeds were chosen solely to require the agent to search for goals within a 140-degree arc.

F.1 Environment 1

Environment 1 encompasses 107 tasks, illustrated in Figure 7, representing all seeds within this arc in the seed range of 0 to 300. The number of hazards is limited to 9, while only one vase exists.
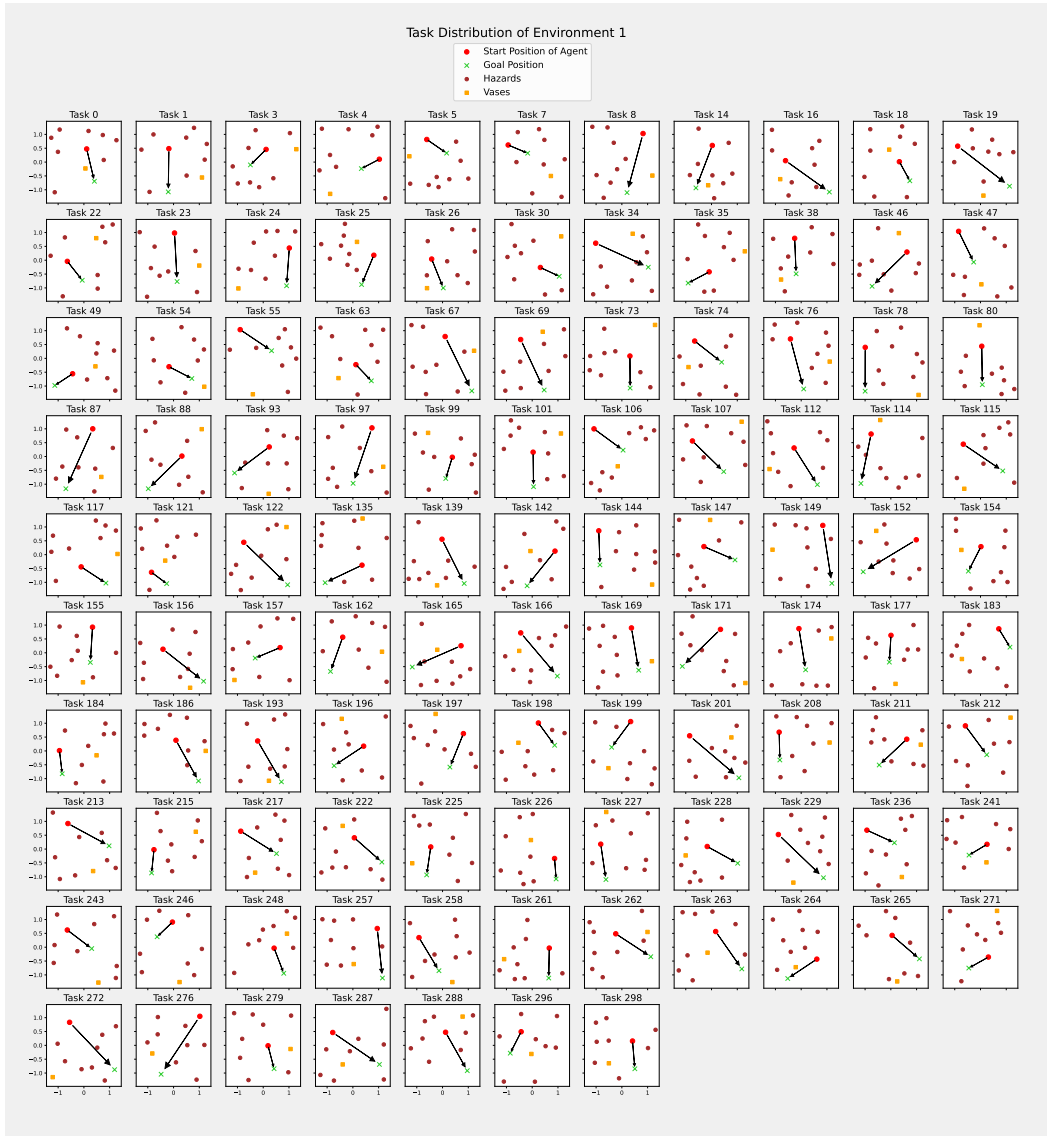

Figure 7: All 107 seeds of the task distribution under Environment 1 are depicted. The tasks were drawn uniformly without replacement.

F.2 Environment 2

Environment 2 consists of 106 tasks, illustrated in Figure 8, encompassing all seeds within the arc for the seed range of 0 to 300. The number of hazards and vases has now increased to ten each, obstructing a direct line of sight between the starting position and the goal in most cases due to several obstacles.
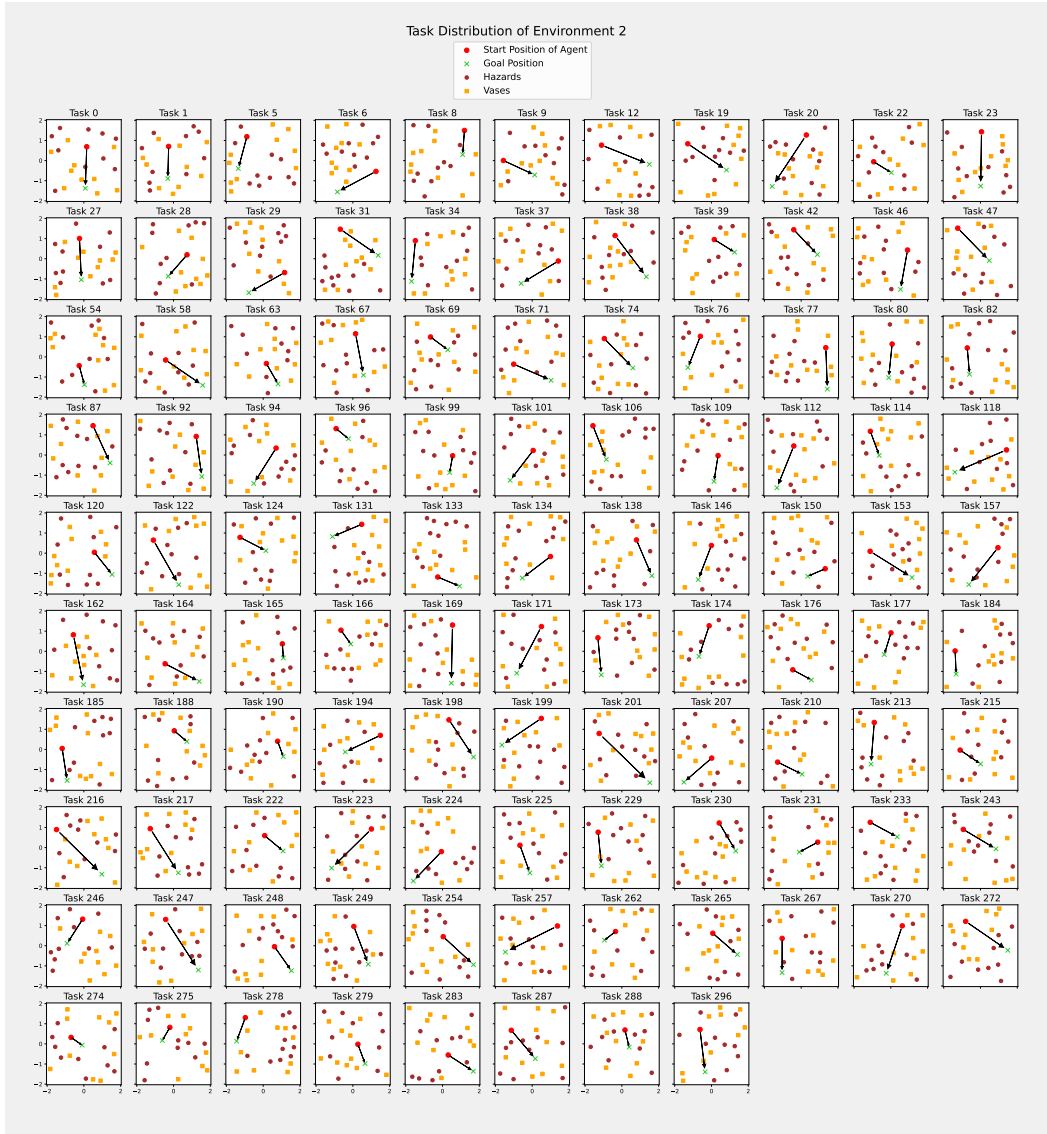
Figure 8: Illustrated are all 106 seeds comprising the task distribution within Environment 2. The tasks were uniformly drawn without replacement.

G   Fine-Tuning

This section clarifies the fine-tuning process for two specific tasks, namely task 122 and task 233, which are part of the broader task distribution denoted by T. The fine-tuning process was meticulously examined using three distinct policy initializations to ensure a comprehensive understanding of the dynamics involved. The initializations are as follows:

1. Random Initialization: This approach simulates the commencement of training from a foundational level, devoid of any prior learning or optimization.

2. Pre-Optimized Policy Using TRPOLag: Here, the policy has been previously fine-tuned with the Trust Region Policy Optimization-Lagrangian (TRPOLag) algorithm, allowing us to assess the impact of prior optimization on the fine-tuning process.

3. C-MAML Initialization with TRPOLag: This method involves initializing the policy through C-MAML, with TRPOLag being applied within the inner optimization loop, offering a unique perspective on the adaptability and efficiency of meta learning strategies in conjunction with TRPOLag.
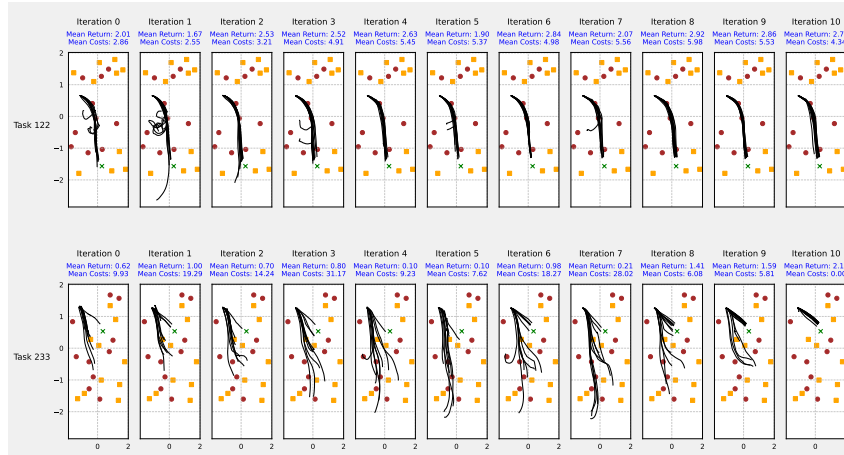
(a) Rollout of the constrained meta-policy
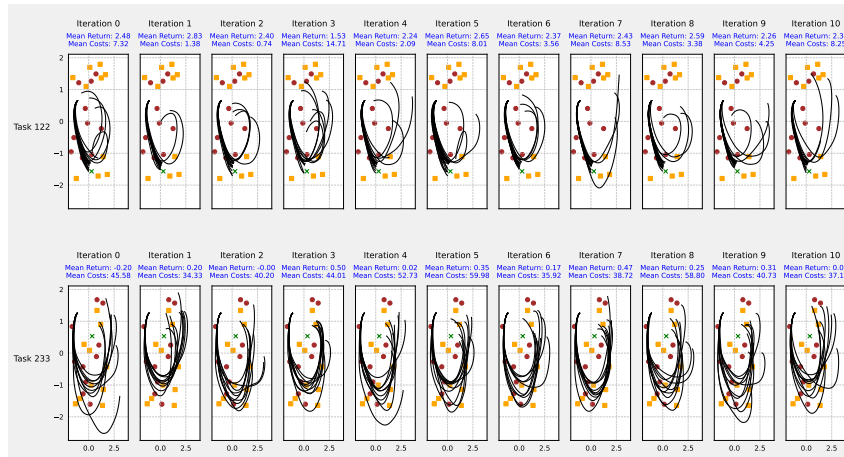
(b) Rollout of a pretrained policy
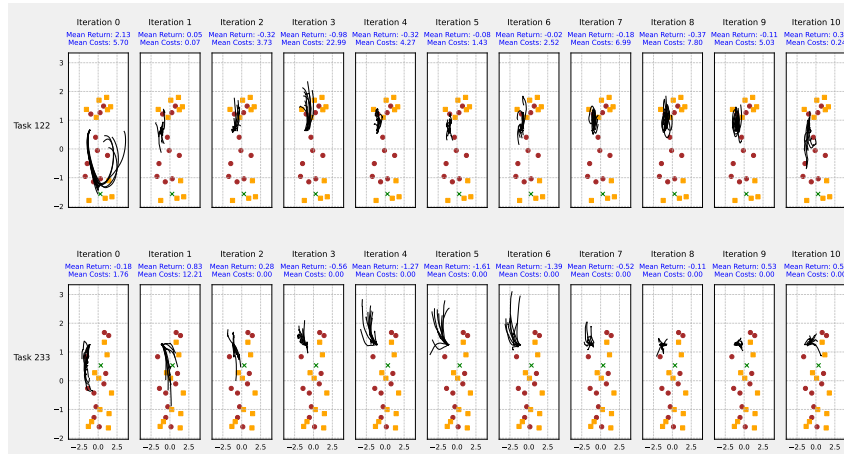
(c) Rollout of a randomly initialized policy

Figure 9: The figure illustrates the behavior of a meta-policy trained with TRPOLag, a pretrained policy, and a randomly initialized policy during the fine-tuning process on two randomly selected tasks from the task distribution.
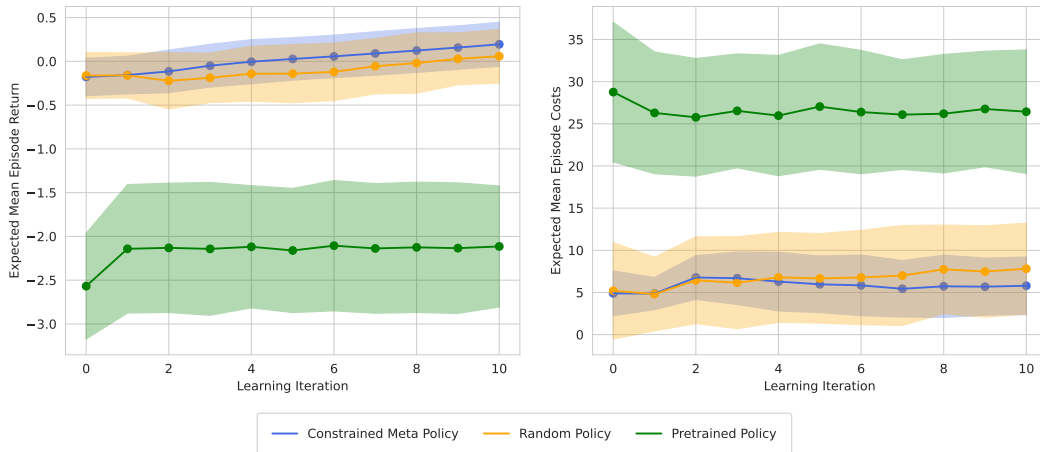
20

H Additional Results of C-MAML TRPOLag

Figure 10: Progression of the mean episode returns and mean episode costs throughout the fine-tuning process. In contrast to Figure 4, an average has been computed across the means of three different seeds, along with the indication of the mean standard deviation of these three seeds. The constrained meta-policy has undergone meta-training using TRPOLag. In fine-tuning, all policies have been adjusted with TRPOLag.

I Additional Results of C-MAML CPO

Figure 11: Progression of the mean episode returns and mean episode costs throughout the fine-tuning process. In contrast to Figure 5, an average has been computed across the means of three different seeds, along with the indication of the mean standard deviation of these three seeds. The constrained meta-policy has undergone meta-training using CPO. In fine-tuning, all policies have been adjusted with CPO.

J Impact of Difficulty Level

The transition from Environment 2 to Environment 1 allows for the modulation of difficulty levels, as the latter incorporates only ten obstacles, in contrast to the 20 obstacles depicted in Figure 12.
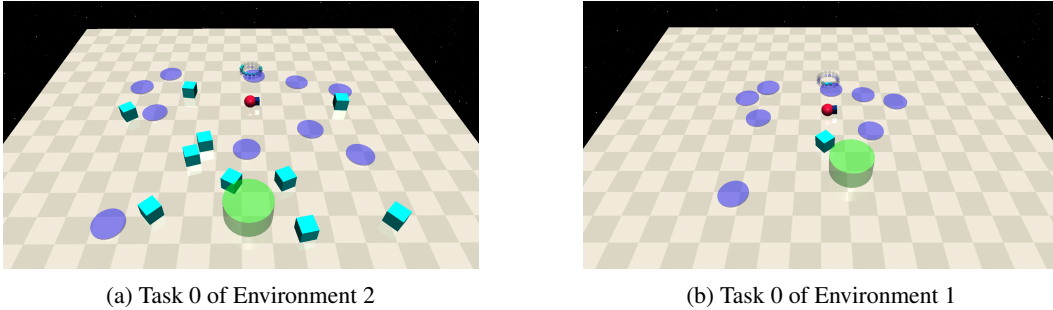

(a) Task 0 of Environment 2

(b) Task 0 of Environment 1

Figure 12: In the left illustration, Environment 2 consists of precisely ten hazards and ten vases in all tasks. Environment 1 mitigates the task's difficulty level by featuring only ten hazards and a single vase. Due to the reduced number of obstacles along the path from the agent to the goal, it is anticipated that the task becomes easier to solve.

The reduction in environmental complexity is theoretically expected to facilitate the achievement of an agent's objectives while incurring lower episode costs. In order to assess the impact of varying difficulty levels on performance, we conducted additional experiments utilizing C-MAML with TRPOLag in Environment 1. The outcomes of these experiments are presented in Figure 13. The results reveal a notably quicker adaptation of the meta-policy compared to both the randomly initialized and pretrained policies. The pretrained policy fails to exhibit improvement after the initial adaptation step, maintaining a consistently low return level.
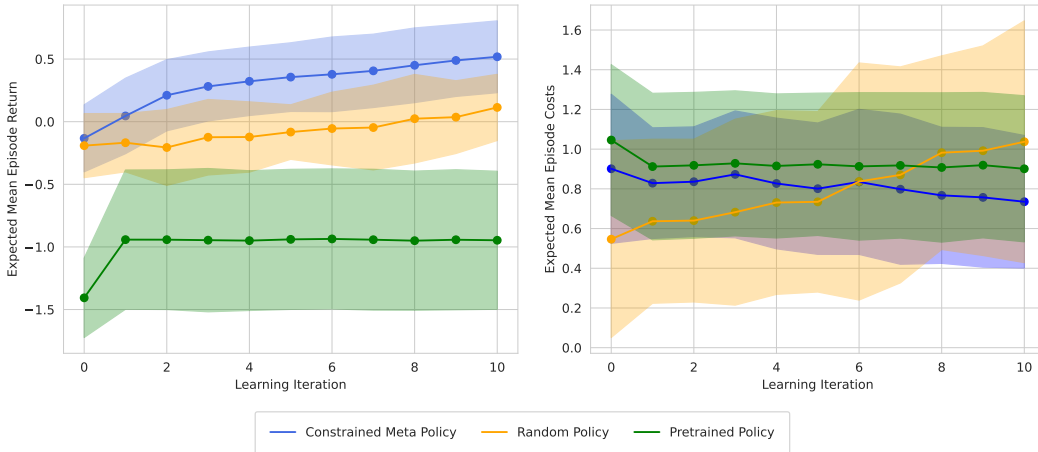

Figure 13: The trajectory of the mean episode return and mean episode costs during the course of fine-tuning on Environment 1 is depicted. As before, averaging was conducted across all tasks within the task distribution.

For all three policy types, the average episode costs exhibit a decrease during fine-tuning in both the meta-policy and the pretrained policy. After ten adaptation steps, the meta-policy reaches the cost limit of $d = 0.75$. In contrast, the random policy begins exploration during fine-tuning, resulting in a sharp increase in mean episode costs.

These results emphasize the robustness of the meta-policy in response to changes in the difficulty level of tasks and the environment.