Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Adaptive Smooth Tchebycheff Attention for Multi-Objective Policy Optimization

Alejandro Murillo-González, Mahmoud Ali and Lantao Liu
Indiana University–Bloomington
{almuri, alimaa, lantao}@iu.edu

***Abstract*—Multi-objective reinforcement learning in robotic domains requires balancing complex, non-convex trade-offs between conflicting objectives. While linear scalarization methods provide stability, they are theoretically incapable of recovering solutions within non-convex regions of the Pareto front. Conversely, static non-linear scalarizations (e.g., Tchebycheff) can theoretically access these regions but often suffer from severe gradient variance and optimization instability in deep RL. In this work, we propose an Adaptive Smooth Tchebycheff framework that resolves this tension by dynamically modulating the curvature of the optimization landscape. We introduce a novel *conflict-driven controller* that regulates the optimization smoothness based on real-time gradient interference. This allows the agent to anneal toward precise, non-convex scalarization when objectives align, while elastically reverting to stable, smooth approximations when destructive gradient conflicts emerge. We validate our approach on a challenging robotic stealth visual search task—a proxy for monitoring of protected/fragile ecosystems—where an agent must balance search, exposure/interference minimization and exploration speed. Extensive ablations confirm that our conflict-aware adaptation enables the robust discovery of Pareto-optimal policies in non-convex regions inaccessible to linear baselines and unstable for static non-linear methods.**

Website: **https://alejandromllo.github.io/research/pasta/.**

***Index Terms*—Robot Learning; Multi-Objective RL (MORL); Multi-Objective Optimization (MOO); Visual Search.**

I. Introduction

Deep reinforcement learning (RL) has demonstrated remarkable efficacy in difficult continuous robot control tasks, including quadrupedal locomotion, unstructured terrain navigation, and dexterous manipulation [26, 21, 32, 34]. Standard approaches typically rely on a scalar reward signal to guide policy optimization. In this paradigm, the environment provides observations and a single scalar reward at each timestep, aggregating various performance metrics—such as energy efficiency, safety, and task completion—into a monolithic objective. This scalar reward often obscures the underlying trade-offs inherent in complex robotic settings and how the preference for each component varies.

Multi-Objective Reinforcement Learning (MORL) addresses scenarios where the environment returns a vector of conflicting rewards [40, 17]. This formulation is critical in robotics, where agents must balance disparate goals—for instance, maximizing speed while minimizing mechanical wear and battery consumption, or navigating to a target while maintaining safety boundaries. The goal of MORL is to find solutions in the Pareto front, the set of policies where no objective can be improved without degrading another.

A common strategy to solve MORL problems is Linear Scalarization (LS), which reduces the reward vector to a scalar via a convex combination of weights. While computationally efficient, LS is theoretically limited: it can only recover solutions on the convex hull of the Pareto front [7, 33]. In domains with non-convex trade-off surfaces—common in complex dynamics or sparse reward settings—LS fails to discover optimal policies in the concave regions of the front, potentially leading to suboptimal or unsafe behaviors that fail to satisfy mission requirements and constraints.

To overcome the geometric limitations of LS, Tchebycheff (TCH) scalarization is frequently employed in the multi-objective optimization (MOO) literature [3, 16]. TCH minimizes the maximum weighted deviation from an ideal "utopia" point, theoretically allowing it to recover all Pareto-optimal solutions regardless of convexity. However, the TCH formulation relies on a *non-differentiable* $\max$ operator. In the context of deep RL, where policy updates rely on backpropagation of gradients, this non-smoothness introduces significant instability [27]. It results in sparse or undefined gradients (subgradients) that hinder the training dynamics [14, 45, 28].

Recently, Lin et al. [27] introduced Smooth Tchebycheff (STCH) scalarization for gradient-based MOO, replacing the non-differentiable $\max$ operator with a Log-Sum-Exp approximation. This formulation is controlled by a strictly positive smoothness parameter, μ. As μ goes to zero, STCH approaches the true Tchebycheff function; however, this creates a bottleneck for gradient flow, directing feedback almost exclusively from the dominant objective (i.e., the one furthest from the "utopia" point). Conversely, as μ moves away from zero, the function behaves increasingly like a linear weighted sum, ensuring smooth gradient flow. This introduces a fundamental trade-off: while the smooth gradient flow mitigates the vanishing gradient problem common in deep learning [35, 18], it comes at the cost of TCH approximation accuracy. Excessive smoothing degrades the TCH precision, effectively reverting the method to a linear scalarization that is incapable of recovering solutions in non-convex regions of the Pareto front.

Furthermore, applying STCH to RL introduces a unique challenge: parameter sensitivity. A fixed μ is difficult to tune because the "dataset" in RL (the agent's experience) is non-stationary. As the policy evolves, the scale of advantages and the proximity to the varying utopia point shift, effectively changing the optimization landscape. A μ that is too small

may cause vanishing gradients, while a large μ fails to exploit the Tchebycheff properties.

In this work, we propose **PASTA** (*Policy-optimization via Adaptive Smooth Tchebycheff Attention*), a novel multi-objective reinforcement learning algorithm specifically designed to address the fundamental tension between optimization stability and geometric precision in model-free RL. At the core of PASTA is our *Adaptive Smooth Tchebycheff* (**ASTCH**) scalarization framework, which dynamically modulates the curvature of the optimization landscape via μ. This integration enables agents to stably recover Pareto-optimal solutions within non-convex concavities—regions theoretically unreachable by linear scalarization—while avoiding the catastrophic gradient variance and feature starvation that plague static non-linear approximations. Within ASTCH, we introduce a novel *conflict-driven controller* that adjusts the smoothing parameter μ based on real-time gradient interference. Our method directly monitors the *conflict ratio* between objective gradients; the controller anneals μ to sharpen the scalarization for precision as training progresses, while employing a *dynamic braking mechanism* to momentarily relax smoothness whenever destructive gradient conflicts threaten the policy update. This "elastic" adaptation prevents gradient starvation and catastrophic forgetting, promoting robust convergence across the entire Pareto front.

Overall, our contributions are summarized as follows:

- We propose the ASTCH scalarization framework. At its core is a novel Adaptive Smoothness Controller that dynamically adjusts the STCH smoothness parameter μ based on an online estimate of inter-objective gradient conflict, effectively eliminating the need for manual tuning and preventing gradient starvation.
- We develop PASTA, an algorithm that integrates the ASTCH framework into an on-policy policy-gradient method. Our formulation explicitly accounts for the non-stationarity of utopia points and value estimates arising in multi-objective RL, and introduces design choices that further improve performance.
- We validate our approach on challenging real-world multi-objective exploration tasks where a ground robot must navigate and visually search an environment while minimizing exposure—a proxy for applications such as stealth monitoring in protected ecosystems or search-and-rescue in contaminated zones. We further demonstrate cross-platform real-world applicability by deploying PASTA on quadrotors tasked with safe traversal among stochastic dynamic obstacles. Finally, we establish broader baseline comparisons using multi-objective MuJoCo benchmarks [12].

II. Related Work

Multi-objective planning and control in robotics involves conflicting objectives, such as energy efficiency versus speed, searching while minimizing interference, or tracking precision versus safety constraints, among many others. We have seen this in diverse problems across RL [51, 59, 63, 31, 22, 30, 25] and control settings [23, 24, 19, 5, 56, 50]. In this work, we are interested in RL problems with multiple objectives, for which we want to find an optimal policy for a specific trade-off, instead of learning the whole Pareto front.

MORL extends the standard Markov Decision Process (MDP) framework to vector-valued rewards, necessitating a trade-off between conflicting objectives [13]. Coverage-based methods use evolutionary algorithms or envelope Q-learning to learn the entire Pareto front [60, 52, 39, 6]. On the other hand, preference-based approaches like ours have relied on scalarization, surrogate utility functions, uncertainty modeling or intricate optimizations methods [29, 61, 57, 58].

Closer to our work leveraging TCH/STCH scalarization, Van Moffaert et al. [53] proposed a TCH action selection strategy that was shown to outperform linear-based approaches. They also designed a multi-objective Q-learning framework that can leverage both linear and non-linear scalarization methods. Conversely, we directly learn a policy by optimizing an objective function that involves a value learned using STCH-prioritized components.

Recently, Qiu et al. [37] reformulated TCH into a min-max-max optimization problem. This structure enables a preference-free framework based on an Upper Confidence Bound (UCB) that learns Pareto optimal policies with provable sample efficiency. They further extend their method to use STCH. For multi-objective multi-agent RL (MO-MARL), Hu et al. [20] proposed a factorized value decomposition method inspired by TCH to minimize the maximum deviation in action-value between the team's global action and the local optimal action of each individual agent. To address the limitations of static local scalarization in MO-MARL, Van Moffaert et al. [54] propose a hybrid adaptive weight algorithm combining the layered exploration of RA-TPLS [10] in early iterations with the adaptive norm-based search intensification of AN-TPLS [10]. This approach removes the reliance on rigid dichotomic schemes and biased seeds, resulting in improved system-wide Pareto coverage. In contrast to the aforementioned approaches, our work focuses on tightly integrating STCH into well-known on-policy RL algorithms to learn preference-optimal policies.

Also on the adaptive side, Abels et al. [1] focus on the setting where scalarization parameters vary over time, and propose a multi-objective Q-network architecture that takes the relative importance of objectives as an input to output weight-dependent Q-values. To mitigate the non-stationarity inherent in changing preferences and to improve sample efficiency in high-dimensional environments, they introduce Diverse Experience Replay (DER), which ensures a diverse range of weight configurations are retained to reduce replay buffer bias. Similar adaptive scalarization parameter schemes have been explored for robotic arm control [46]. Rather than selecting scalarization preferences, we focus on balancing optimization smoothness, STCH approximation precision and solution quality.

III. PRELIMINARIES

A. Multi-Objective Optimization (MOO)

Many tasks involve optimizing multiple, often conflicting, objectives simultaneously. This gives rise to a Multi-objective Optimization Problem (MOP), which is formally defined as:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbf{F}(\mathbf{x}) = \min_{\mathbf{x}\in\mathcal{X}} [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})]^\top \tag{1}$$

where $\mathbf{x}$ is the decision vector (e.g., policy parameters θ) from the feasible set $\mathcal{X}$, and $\mathbf{F} : \mathcal{X} \to \mathbb{R}^m$ is the vector-valued objective function mapping $\mathbf{x}$ to an m-dimensional objective.

1) Pareto Optimality: MOPs do not typically have a single optimal solution. Instead, we seek a set of "best" trade-offs, defined by the concept of dominance.

Dominance: A solution $\mathbf{x}_A$ **dominates** $\mathbf{x}_B$ (denoted $\mathbf{x}_A \prec \mathbf{x}_B$) if its objective vector $\mathbf{u} = \mathbf{F}(\mathbf{x}_A)$ is no worse in any objective and strictly better in at least one:

$$\forall i \in \{1, \ldots, m\} : u_i \leq v_i \quad \wedge \quad \exists j \in \{1, \ldots, m\} : u_j < v_j$$

where $\mathbf{v} = \mathbf{F}(\mathbf{x}_B)$.

The concept of dominance lets us define additional key components of the MOP. A solution $\mathbf{x}^* \in \mathcal{X}$ is **Pareto optimal** if no other solution $\mathbf{x} \in \mathcal{X}$ dominates it. The set of all Pareto optimal solutions in the decision space is the **Pareto Set**, $\mathcal{PS}$. Its image in the objective space, $\mathcal{PF} = \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{PS}\}$, is the **Pareto Front**. Finally, the goal of MOO is to find a set of solutions that approximates the $\mathcal{PF}$.

2) Solution Methods: We focus on *gradient-based* MOO solution methods, given their compatibility with deep learning based RL. Most gradient-based MOO methods can be categorized as *scalarization* or *adaptive gradient* approaches [27]. The latter seek to simultaneously improve all objectives every iteration by discovering an appropriate gradient direction [41, 8]. Scalarization-based approaches convert the MOP into a parameterized Single-Objective Problem (SOP).

Linear Scalarization: The simplest scalarization method is the weighted sum: $L(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x})$, for a weight vector $\mathbf{w}$ with $w_i \geq 0, \sum w_i = 1$. While simple, this method can *only* find solutions on the *convex hull* of the Pareto Front [7]. It fails to find solutions in non-convex regions, which are necessary and common in practice [11].

Tchebycheff Scalarization: To address this, Tchebycheff scalarization is widely used [3]. It works by minimizing the maximum weighted distance to an ideal "utopia" point $\mathbf{z}^* = [z_1^*, \ldots, z_m^*]^\top$. The utopia point consists of components z_i^* that represent the theoretical minimum (or slightly below) of each objective f_i individually. Crucially, $\mathbf{z}^*$ is strictly infeasible; achieving all ideal values simultaneously is impossible due to the conflicting nature of the objectives. Conceptually, this acts like a "carrot on a stick" for the optimizer: the utopia point is the carrot suspended just beyond the feasible region. The optimizer is driven to minimize the gap to this unreachable target, which effectively pulls the solution as far as possible towards the Pareto frontier. The function to minimize is:

$$L_T(\mathbf{x}, \mathbf{w}, \mathbf{z}^*) = \max_{i\in\{1,\ldots,m\}} \{w_i(f_i(\mathbf{x}) - z_i^*)\}. \tag{2}$$

The Tchebycheff method is guaranteed to find *any* Pareto optimal solution [47, 27], regardless of the front's shape.

Gradient-Based Tchebycheff Optimization: The nonlinear $\max$ function in Eq. (2) poses a challenge for gradient-based methods. In practice, the subgradient is used [28]. The gradient of L_T with respect to $\mathbf{x}$ (or policy parameters θ) is the gradient of the *single objective* that is currently the "worst-performer" (the one that determines the $\max$):

$$j = \arg\max_i \{w_i(f_i(\mathbf{x}) - z_i^*)\} \implies \nabla_{\mathbf{x}} L_T \approx \nabla_{\mathbf{x}} \big(w_j f_j(\mathbf{x})\big)$$

This approach introduces its own challenges. The subgradient can "jump" abruptly as the $\arg\max$ switches between objectives, leading to unstable training (see Fig. 2 of [27] for an example). It also requires a well-defined utopia point $\mathbf{z}^*$, which may not be known a priori and would need to be estimated. This process is formalized in Alg. 1 (Appendix A).

B. Multi-Objective Reinforcement Learning (MORL)

MORL models the problem as a Multi-Objective Markov Decision Process (MOMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, \mathbf{r}, \gamma)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the state transition dynamics. Unlike standard MDPs, the reward function $\mathbf{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ returns a vector of m distinct objective signals $\mathbf{r}(s, a) = [r_1(s, a), \ldots, r_m(s, a)]^\top$.

The goal of the agent is to learn a policy $\pi(a|s)$ that optimizes the expected discounted vector return: $\mathbf{J}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t)\right]$ where $\gamma \in [0, 1)$ is the discount factor.

Since objectives are often conflicting, no single set of *policy parameters* maximizes all simultaneously. We therefore seek policies that are *Pareto-optimal* [17]. A policy π is said to *dominate* another policy π' (denoted $\mathbf{J}(\pi) \succ \mathbf{J}(\pi')$) if its **performance** is superior or equal in all objectives and strictly superior in at least one:

$$\forall i \in \{1, \ldots, m\}, J_i(\pi) \geq J_i(\pi') \ \wedge \ \exists j, J_j(\pi) > J_j(\pi'). \tag{3}$$

The set of non-dominated policies constitutes the *Pareto Front*.

C. Policy Optimization Algorithms

This section draws from [2, 49, 42, 44] to trace the evolution of modern policy optimization algorithms.

1) Policy Gradient Methods (PG) [49]: directly optimize the parameters θ of a policy π_θ to maximize the expected return. This is typically achieved via stochastic gradient ascent on the policy's performance $J(\pi_{\theta_k})$:

$$\theta_{k+1} = \theta_k + \alpha \nabla_{\theta_k} J(\pi_{\theta_k}), \tag{4}$$

where α is the learning rate and the policy gradient is:

$$\nabla_{\theta_k} J(\pi_{\theta_k}) = \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_k}} \left[\sum_{t=0}^{T} \nabla_{\theta_k} \log \pi_{\theta_k}(a \mid s) A^{\pi_{\theta_k}}(s, a) \right]. \tag{5}$$

Here, $A^{\pi_{\theta_k}}(s, a)$ is the *advantage function*, often derived from a simultaneously learned value function V_{ϕ_k}.

However, PG methods are sensitive to the learning rate α. A single step that is too large can drastically change the policy, leading to a "performance collapse".

2) Trust Region Policy Optimization (TRPO) [42]: tries to address the instability of PG by taking the largest possible step that improves performance while explicitly constraining how much the new policy π_θ can diverge from the old one π_{θ_k}. This "trust region" is enforced using the Kullback-Leibler (KL) divergence. The theoretical TRPO update is a constrained optimization problem:

$$\theta_{k+1} = \arg\max_{\theta} L(\theta, \theta_k) \;\; \texttt{s.t.} \;\; \bar{D}_{KL}(\theta \,||\, \theta_k) \leq \psi \quad (6)$$

where ψ is the step size (the "size" of the trust region), $L(\theta, \theta_k)$ is the *surrogate advantage*:

$$L(\theta, \theta_k) = \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_k}} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_k}(a \mid s)} A^{\pi_{\theta_k}}(s,a) \right], \quad (7)$$

and $\bar{D}_{KL}(\theta \;||\; \theta_k)$ is the average KL-divergence between policies, both computed using states visited and corresponding actions sampled by the old policy. In practice, the authors solve an approximate optimization problem derived from a Taylor expansion of Eq. (6).

3) Proximal Policy Optimization (PPO) [44]: emerged as an alternative that retains the stability of TRPO but with a simpler, first-order optimization approach. PPO's key insight is to reformulate the objective function to penalize large policy changes directly, rather than using a hard KL-divergence constraint. PPO implements this via a *clipped surrogate* objective function. Specifically, the full objective comprises three components. First, the *clipped surrogate* objective penalizes large policy deviations. Defining the probability ratio $r_t(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}$, this loss is given by: $L_t^{CLIP}(\theta) =$

$$\min \Big(r_t(\theta) A^{\pi_{\theta_k}}(s,a), \texttt{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big) A^{\pi_{\theta_k}}(s,a) \Big), \quad (8)$$

where ϵ is a hyperparameter bounding the update. Second, a value function loss $L_t^{VF}(\phi) = (V_\phi(s_t) - V_t^{\text{target}})^2$ ensures accurate advantage estimation. Third, an entropy bonus $S[\pi_\theta](s_t)$ encourages exploration. The final parameters are updated by maximizing the weighted combination of these terms:

$$\theta_{k+1}, \phi_{k+1} = \quad (9)$$
$$\arg\max_{\theta,\phi} \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_k}} \Big[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\phi) + c_2 S[\pi_\theta](s_t) \Big],$$

where c_1 and c_2 are coefficients weighting the value error and entropy bonus, respectively. See Appendix B for additional details and pseudocode.

IV. Policy Optimization via Smooth Tchebycheff Attention

We address the challenge of learning Pareto-optimal policies in non-convex MORL domains, where standard linear scalarization of the reward fails to recover solutions beyond the convex hull. To resolve these complex trade-offs, we integrate *Smooth Tchebycheff (STCH) scalarization* [27] into an on-policy policy-gradient framework based on PPO [44]. Specifically, integrating STCH seeks to unlock the discovery of non-convex solutions while improving the gradient stability essential for deep RL.

Crucially, the STCH analytical derivative naturally functions as a principled *attention mechanism*. This section establishes the core mathematical integration of this insight. However, relying on a static scalarization curvature risks gradient starvation and optimization collapse as the policy evolves. To overcome this, Section V introduces a conflict-driven controller to actively modulate the STCH smoothness. This addition completes the "adaptive" component of our framework, culminating in our full proposed algorithm: **PASTA**.

A. Smooth Tchebycheff Scalarization

We consider m objectives with user-defined preference weights $\mathbf{w} \in \Delta^{m-1}$, such that $w_i \geq 0, \sum w_i = 1$. Classical Tchebycheff scalarization minimizes the maximum weighted deviation from an ideal (utopia) point $\mathbf{z}^*$. In RL, however, optimization is naturally posed as a maximization problem.

We propose to do this by fixing the utopia point at $z_i^* = \zeta > 1$. To ensure scale invariance and a stable reference point, we also dynamically normalize raw *returns* R_i using running estimates of the global minimum $R_i^{\min}$ and maximum $R_i^{\max}$ observed during training: $\bar{r}_i = (R_i - R_i^{\min})/(R_i^{\max} - R_i^{\min} + \epsilon)$, such that $\bar{r}_i \in [0,1]$. As a result, we can turn our MOP into a SOP (scalarize) using STCH [27] in maximization form, with respect to the normalized returns:

$$\mathcal{S}_{\text{STCH}}(\bar{\mathbf{r}}) = -\mu \log \sum_{i=1}^{m} \exp\left(\frac{w_i(z_i^* - \bar{r}_i)}{\mu} \right), \quad (10)$$

where $\mu > 0$ is the smoothness parameter.

As $\mu \to 0$, Eq. (10) converges to the classical (hard) Tchebycheff operator. As $\mu \to \infty$, it approaches the linear weighted sum $\sum_i w_i \bar{r}_i$. Importantly, μ controls not only bias–variance trade-offs but the *curvature of the Pareto front approximation* and, consequently, the structure of the induced gradients. See Appendix C for illustrations and further analysis of STCH.

B. Smooth Tchebycheff Attention

A defining property of STCH is that its gradient induces a normalized, positive weighting over objectives:

$$\delta_i(\bar{\mathbf{r}}) \;\triangleq\; \frac{\partial \mathcal{S}_{\text{STCH}}(\bar{\mathbf{r}})}{\partial \bar{r}_i} = \frac{\exp(w_i(z_i^* - \bar{r}_i)/\mu)}{\sum_j \exp\big(w_j(z_j^* - \bar{r}_j)/\mu\big)}. \quad (11)$$

These coefficients admit a precise interpretation:

- Objectives with larger weighted deviation from the utopia point receive exponentially higher attention;
- μ controls how sharply attention concentrates on the limiting objective, i.e., the one with largest weighted deviation from the utopia.

Crucially, $\boldsymbol{\delta}$ is *not* a heuristic weighting. It is the exact multiplier with which each objective contributes to the scalarized gradient. This observation motivates our design choice, described in the remaining of this text, to propagate STCH attention consistently across policy and value updates.

However, in the low-μ regime, the attention weights in Eq. (11) become sharply peaked. While necessary for resolving non-convex Pareto trade-offs, this concentration induces a well-known failure mode in deep learning: *gradient starvation* [35, 18]. Objectives that are temporarily non-limiting receive near-zero gradient signal, causing the shared representation to forget features critical for those objectives.

To counteract this effect, we introduce a **Maintenance Rate** $\rho \in (0, 1)$, which enforces a minimum gradient budget for all objectives. As such, we define the effective attention as

$$\boldsymbol{\eta} = (1 - \rho)\boldsymbol{\delta} + \rho\mathbf{u}, \qquad \mathbf{u} = (1/m, \ldots, 1/m). \tag{12}$$

This guarantees that each objective receives at least ρ/m of the gradient signal. Conceptually, this mechanism plays a role analogous to persistent excitation in adaptive control: it ensures that the system remains identifiable along all objective dimensions, enabling rapid re-adaptation when trade-offs shift.

C. Attention-Aligned Branched Critic

To ensure that value learning prioritizes the same objectives indicated by STCH, we employ a **Branched Critic** architecture with a shared feature extractor and m independent value heads. The new critic loss, replacing L_t^{VF} in Eq. (9), is:

$$L_t^{VF} = \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{m} \eta_{t,i} \left(V_\phi^{(i)}(s_t) - y_t^{(i)} \right)^2, \tag{13}$$

where $y_t^{(i)}$ denotes the target return for objective i. Consistent with PPO [44], we calculate $y_t^{(i)}$ as the λ-return, using Generalized Advantage Estimation (GAE) [43] to obtain a variance-reduced estimate of the discounted return (Sec. III-B).

This attention-aligned critic ensures that representational capacity is allocated where it most reduces variance in the scalarized advantage. This is also consistent with TCH's focus on bottleneck objectives. Without this alignment, the critic would expend capacity modeling objectives that contribute little to the policy improvement.

D. Vectorized Advantage

Unlike standard scalarized MORL, which collapses returns into a single utility value prior to optimization, we maintain a vector of advantages over which we do gradient projection later on. For each objective $i \in \{1, \ldots, m\}$, we compute the Advantage $A_i(s_t, a_t)$ using GAE. We then apply per-objective normalization over the batch:

$$\hat{A}_i(s, a) = \big(A_i(s, a) - \mathbb{E}[A_i]\big) / \big(\sigma(A_i) + \epsilon\big). \tag{14}$$

This yields a vector advantage $\hat{\mathbf{A}}(s, a) \in \mathbb{R}^m$.

Rather than scalarizing $\hat{\mathbf{A}}$ immediately, which effectively destroys information about gradient *conflict*, we use this vector during the actor's optimization. Specifically, deferring scalarization allows the actor to compute distinct policy gradients $\mathbf{g}_i$ for each objective. This separation is a prerequisite for our optimization strategy, as it exposes the geometric conflicts required to mitigate destructive interference, explained next.

E. Conflict-Aware Gradient Projection (PCGrad)

Despite attention-aligned updates, non-convex Pareto trade-offs can still induce destructive gradient interference. To address this issue, we integrate a subtly modified variant of Projecting Conflicting Gradients (PCGrad) [62] that operates at the level of individual objectives, rather than tasks, enabling conflict projection directly in objective space.

Let $\mathbf{g}_i = \nabla_\theta L_i^{CLIP}$ (recall Eq. (8)) denote the policy gradient associated with objective i via the vectorized advantage $\hat{A}_i(s, a)$ from Eq. (14). For randomly sampled pairs of objectives (i, j), PCGrad detects conflicts via the inner product: $\mathbf{g}_i^\top \mathbf{g}_j < 0$. If a conflict is detected, $\mathbf{g}_i$ is projected onto $\mathbf{g}_j$'s normal plane:

$$\mathbf{g}_i^{\text{proj}} = \mathbf{g}_i - \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j. \tag{15}$$

Finally, replacing the standard scalarized PPO update, we sum the projected gradients of m distinct clipped objectives L_i^{CLIP} to update the policy: $\theta_{k+1} = \theta_k + \alpha_\theta \sum_{i=1}^{m} \mathbf{g}_i^{\text{proj}}$. This ensures that the optimization respects individual trust-region constraints while mitigating geometric conflicts.

Crucially, we note that PCGrad also provides a principled measure of gradient interference. Based on this we define the **Conflict Ratio** κ_t at iteration t as:

$$\kappa_t = \frac{\#\{(i, j) : \mathbf{g}_i^\top \mathbf{g}_j < 0\}}{\#\{(i, j)\}}, \tag{16}$$

i.e., the fraction of sampled gradient pairs exhibiting conflict.

A key idea of this work is to *elevate* κ_t from a passive diagnostic to an active control signal. We show that κ_t provides a reliable, real-time estimate of the agent's ability to reconcile the current Pareto trade-offs, and we leverage it as the feedback variable in our adaptive smoothness controller, closing the loop between optimization dynamics and scalarization curvature.

V. Conflict-Driven Adaptive Smoothness

We introduce a mechanism to adapt the smoothness parameter μ of STCH during training. The controller leverages objective-gradient interference, quantified by the conflict ratio in Eq. (16), as a real-time feedback signal. This feedback allows the scalarization smoothness to adjust dynamically in response to the evolving optimization landscape.

A. Adaptive Smoothness Mechanism

The controller is designed as a closed-loop regulator that varies μ to maintain a "healthy" level of gradient conflict. It consists of three components: a monotonic decay baseline, a conflict-based boost (brake), and an inertial smoother. Intuitively, this creates an "elastic" optimization landscape: the monotonic decay progressively sharpens the scalarization to recover precise solutions within non-convex concavities. Simultaneously, the dynamic brake momentarily relaxes this curvature upon high gradient interference, permitting the agent to traverse scenarios that would otherwise stall convergence.

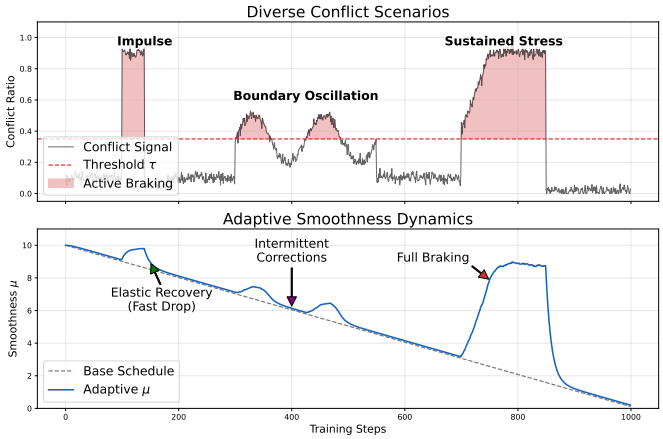

Fig. 1: The conflict ratio κ_t (top), computed from inter-objective gradient interference, serves as a feedback signal for adapting the Smooth Tchebycheff parameter μ_t (bottom). Transient spikes in conflict trigger rapid increases in μ_t ("elastic recovery") to stabilize optimization, while sustained high conflict induces prolonged braking toward a linear-like scalarization. As conflict subsides, μ_t decays back toward the base schedule, enabling annealing toward sharper non-convex scalarization.

1) Base Decay Schedule: We define a base schedule $\mu_{base}(t)$ that linearly anneals from a stable starting value μ_{start} to a precise terminal value μ_{min} over the course of training steps T:

$$\mu_{base}(t) = \mu_{start} - (\mu_{start} - \mu_{min}) \times \min\left(1, \frac{t}{T}\right). \quad (17)$$

This provides the global pressure to converge toward hard Tchebycheff scalarization.

2) Conflict-Driven Braking: Another core component is the dynamic braking mechanism. We define a **Conflict Threshold** τ, representing the maximum acceptable ratio of conflicting gradients. At each step t, we observe the conflict ratio κ_t from the PCGrad update. If $\kappa_t > \tau$, the optimization landscape is deemed challenging for the current policy stability. The controller calculates a boost factor proportional to the excess conflict:

$$\beta_{boost} = \begin{cases} \frac{\kappa_t - \tau}{1 - \tau} & \text{if } \kappa_t > \tau, \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

The target smoothness μ_t^* is then interpolated between the base schedule and the maximum stability value μ_{max}:

$$\mu_t^* = \mu_{base}(t) + \beta_{boost} \cdot (\mu_{max} - \mu_{base}(t)). \quad (19)$$

This mechanism effectively "brakes" the annealing process, reverting to a more linear-like (convex) formulation when gradient conflicts threaten to destabilize the update.

3) Inertial Smoothing: To prevent rapid oscillations in the objective function, which could violate the stationary distribution assumption of PPO, we apply an Exponential Moving Average (EMA) to the controller's output. The final operational μ_t is updated as:

$$\mu_t = (1 - \lambda)\mu_{t-1} + \lambda \mu_t^*. \quad (20)$$

where λ is a small smoothing factor.

B. Controller Dynamics

As shown in Fig. 1, this control law generates distinct behaviors we found to be critical for adaptive scalarization:

- *Elastic Recovery:* In response to transient spikes in conflict (e.g., entering a new area of the state space), μ rapidly increases to stabilize the gradients, then decays elastically back to the base schedule as the agent adapts.
- *Sustained Braking:* If the agent encounters a fundamentally high-conflict trade-off (a "wall"), the controller maintains a high μ, forcing a linear-like approximation that allows the agent to bypass the local minima before attempting to resolve the non-convexity again.

The complete integration seeks that the scalarization curvature (μ) matches the agent's current ability to resolve gradient conflicts, enabling convergence to the non-convex Pareto front.

See Appendix D for *pseudocode* (Alg. 3) describing the integration of our multi-objective policy optimization algorithm with the adaptive STCH smoothness.

VI. Results

We evaluate the proposed PASTA algorithm across a diverse suite of continuous multi-objective environments, encompassing ground-based stealth visual search, quadrotor navigation among dynamic obstacles, and high-dimensional locomotion tasks. Our analysis focuses on three key dimensions: **(1)** the ability to recover optimal policies on non-convex Pareto fronts, **(2)** the sample efficiency improvement, and **(3)** the robustness of the method to architectural choices.

Notably, PASTA *uses the same hyperparameters* across all experiments and environments (see Appendix E1), highlighting that our results do not depend on case-specific fine-tuning.

A. Baselines

We benchmark PASTA against diverse scalarization approaches using identical PPO architectures (see Appendix E2) to ensure a fair comparison. These baselines include: **Linear Scalarization**, which is fundamentally just the standard PPO algorithm, but with the scalarization step (dynamically normalizing and collapsing the reward components into a weighted sum) taking place inside the algorithm rather than within the RL environment—a formulation that inherently fails to explore concave regions of the Pareto front; **Tchebycheff Scalarization**, which can recover the full front but suffers from gradient starvation and instability; and **Fixed Smooth Tchebycheff (STCH)** across a spectrum of smoothing parameters $\mu \in \{0.01, 0.1, 0.5, 1.0, 5.0, 10.0\}$. The latter serves to verify if PASTA effectively automates the trade-off search between approximation error and optimization smoothness.

B. Performance Metrics

We evaluate convergence, diversity, and robustness using a suite of complementary metrics. Our primary measure is **Hypervolume** (HV), which captures both convergence to the Pareto front and solution diversity via the dominated objective-space volume. To assess reliability across preferences, we

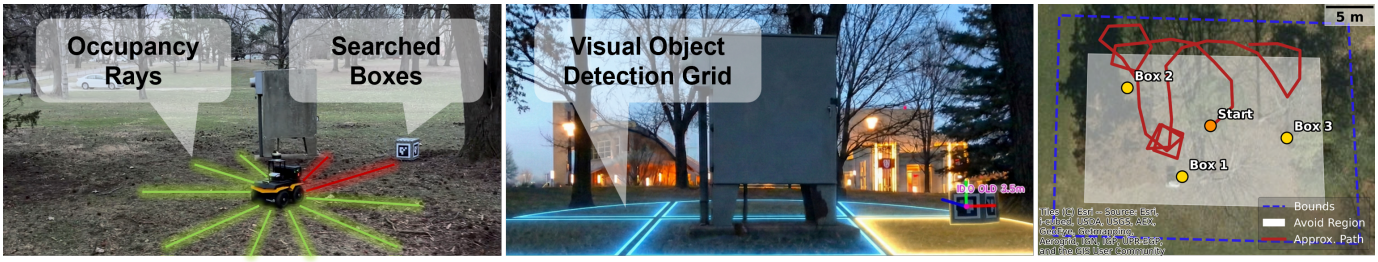

Fig. 2: Real-world Stealth Visual Search Experiments. *(Left)* Task setup showing the UGV, searched items and the occupancy ray observation space derived from LiDAR. *(Center)* Onboard camera view showing the grid-representation of the object detection pipeline. *(Right)* Trajectory aerial view. The robot path (red) minimizes intrusion into the "Avoid Region" (white), entering only as necessary to localize the targets.

report the **Target Dominance Rate** (Win Rate), which measures the fraction of evaluated preferences for which a method attains the highest mean HV, and the **Objective Dominance Rate**, which quantifies how frequently individual objectives are maximized without excessive trade-offs, i.e., overly sacrificing secondary objectives. To aggregate performance across seeds and environments while mitigating outlier effects, we use **Dolan-Moré Performance Profiles** (DMP), reporting the Area Under the Curve (**AUC**) as a robustness indicator. For completeness, we additionally report **Expected Utility** (EU), noting that it may be biased by differing reward scales, rendering it potentially misleading in non-convex MOPs. Full metric definitions are provided in Appendix F.

C. Evaluation in Stealth Visual Search

We evaluate the PASTA algorithm in the real-world with a differential-drive ground-robot traversing unstructured terrain. The robot is tasked with searching for randomly scattered objects while simultaneously minimizing its exposure. This environment serves as a proxy for critical real-world deployments, such as an autonomous rover monitoring endangered species in a fragile habitat or a search-and-rescue robot navigating a radiologically contaminated disaster zone. Specifically, this mission requires the agent to dynamically balance three conflicting objectives: (1) *Visual Object Detection* (maximizing the number of unique targets found), (2) *Stealth* (minimizing exposure time to prevent ecosystem disruption or radiation contamination), and (3) *Exploration* (ensuring thorough workspace coverage).

The robot operates within a bounded arena, defined by $[-x_{\text{dim}}, x_{\text{dim}}] \times [-y_{\text{dim}}, y_{\text{dim}}]$, and is controlled via a continuous action vector $a_t = [v, \omega]$ that dictates its linear ($v \in [0, 1]$) and angular ($\omega \in [-1, 1]$) velocities. At each time step, the agent receives an observation vector s_t comprising its proprioceptive state—its position (x, y) and heading $(\cos\theta, \sin\theta)$—alongside exteroceptive sensor data. This sensor suite features a flattened 2×3 spatial grid over a 98° field of view to encode detected objects, and a LIDAR point cloud downsampled to 20 rays for obstacle avoidance. The policy was trained in a simulation environment built upon the Gymnasium framework. Further details are provided in Appendix G.

The real-world experimental scenarios are visualized in Fig. 2 and **Video 1**[1]. We validate the agent across *eight* diverse preference vectors $\mathbf{w}$, ranging from balanced policies to extreme specialization. Fig. 3 illustrates the Hypervolume performance, while Table I details the complete statistics across our metrics. Our method, PASTA, significantly outperforms all baselines, achieving a Hypervolume of 0.291, which constitutes a $\mathbf{45.5}\%$ improvement over the strongest baseline (STCH $\mu = 10.0$). This superiority is further evidenced by a $\mathbf{100}\%$ Win Rate and an Objective Dominance Rate of $\mathbf{58.3}\%$, vastly exceeding the next best method (12.5%). Regarding the baselines, we observe interesting behavioral trends in the bar plot: STCH with small smoothness values (e.g., $\mu = 0.01$) performs nearly identically to the standard TCH approach (0.125 vs 0.122 HV). Conversely, as μ increases to 10.0, performance aligns closely with the Linear baseline (0.200 vs 0.199 HV), providing empirical evidence that STCH transitions from Tchebycheff-like to linear-like behavior as the smoothness parameter moves away from zero.

Due to space constraints, we provide a detailed breakdown of results per preference vector (Table VI) and additional training curves (Fig. 8) in Appendix G3. Notably, it also includes radar chart comparisons (Fig. 9) against the strongest baselines. These visualizations confirm the robustness of our approach: in most tested cases, PASTA achieves the highest utility in at least two out of the three optimized objectives, demonstrating superior trade-off management.

TABLE I: Comparison against the baselines. Statistics computed over eight preference vectors across five seeds.

Method	Hypervolume ↑	Win Rate ↑ [%]	Obj. Dom. Rate ↑ [%]	DMP AUC ↑
PASTA (Ours)	**0.291 ± 0.024**	**100.0**	**58.3**	**0.981**
Tchebycheff	0.122 ± 0.060	0.0	0.0	0.325
STCH (0.01)	0.125 ± 0.057	0.0	0.0	0.367
STCH (0.1)	0.140 ± 0.079	0.0	8.3	0.434
STCH (0.5)	0.165 ± 0.070	0.0	8.3	0.520
STCH (1.0)	0.187 ± 0.061	0.0	4.2	0.611
STCH (5.0)	0.192 ± 0.066	0.0	4.2	0.642
STCH (10.0)	0.200 ± 0.050	0.0	12.5	0.700
Linear	0.199 ± 0.066	0.0	4.2	0.678

1) Ablation Studies: Table II summarizes the ablation analysis, validating the contribution of each component in our framework. Appendix H provides more detailed breakdowns.

First, we observe that mitigating *gradient interference* is critical; removing PCGrad results in a drop in most metrics,

[1] **Video 1**: https://youtu.be/xLynYv05Bdc

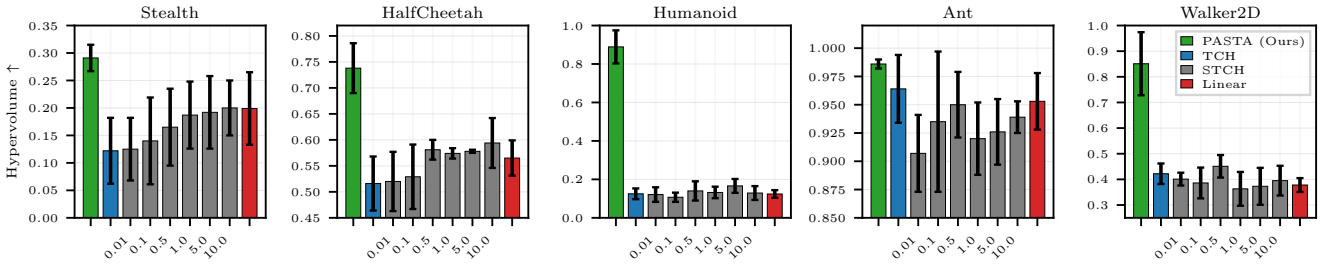

Fig. 3: Performance on Stealth Visual Search and MuJoCo multi-objective environments. Results are aggregated across five seeds.

confirming the usefulness of relying on gradient projection. In the *Controller* analysis, while the "No Conflict" variant achieves a comparable HV to PASTA, our full method maintains a higher Win Rate ($\mathbf{50.0}\%$), suggesting that the conflict signal contributes to robustness. Notably, removing the decay schedule ("No Decay") degrades performance the most, confirming that temporal convergence towards small smoothness values (ideal to closely approximate standard TCH) is key and can be achieved successfully with our controller. Regarding the *Critic*, the results reveal a two-fold insight. First, architecture plays a primary role: the "Branched" baselines significantly outperform the "Shared" variants (**0.277** vs 0.249 average HV), confirming that distinct value heads are necessary to decouple conflicting objective estimates. However, architecture alone is insufficient; our proposed attention-weighted loss provides the decisive advantage. Adding this weighting to the branched architecture (PASTA) amplifies performance, increasing the HV by $\mathbf{10.2}\%$ and boosting the DMP AUC to $\mathbf{0.957}$. This demonstrates that aligning the critic's optimization focus with the actor's scalarization is just as critical as the architectural separation. Finally, the *Smoothness* comparison highlights the benefit of adaptivity. While PASTA matches the *average* performance of the best fixed scalarization ($\mu = 5.0$), the detailed breakdown in Appendix H4 reveals that the latter's success is skewed by specific outliers. In fact, as shown in Table X, for some preference vectors, $\mu = 5.0$ yielded the worst HV performance of all configurations. This underscores that PASTA achieves high performance through robustness rather than chance, effectively automating the otherwise extensive and expensive search for the optimal trade-off surface without the instability and uncertainty inherent to fixed parameters.

2) Sensitivity Analysis: Table III summarizes the sensitivity of our algorithm to its three core hyperparameters. We find that *for all choices of hyperparameters our method still beats all the baselines*, speaking to its robustness. In general, we see a distinct performance region for the Conflict Ratio (κ), indicating that values just below the midpoint of the range are sufficient, while larger values remain viable but suboptimal. For the Moving Average factor (λ), we confirm that a stable, slowly-updating memory of gradient conflict is preferred over reactive signals. Finally, the Maintenance Rate (ρ), shows that while the controller is robust to a wide range of values, a minimum threshold is essential to prevent performance degradation, which we hypothesize arises due to vanishing gradients from STCH. See Appendix I for additional results, including a study of the effect of the utopia point (z_i^*) selection.

TABLE II: Ablation Studies Results. Statistics computed over eight preference vectors across five seeds.

Method	Hypervolume ↑	Win Rate ↑ [%]	Obj. Dom. Rate ↑ [%]	DMP AUC ↑
PCGrad				
PASTA (Ours)	**0.291 ± 0.024**	**87.5**	**58.3**	**0.962**
No PCGrad	0.187 ± 0.065	0.0	12.5	0.598
Weighted PCGrad	0.270 ± 0.019	12.5	29.2	0.914
Controller				
PASTA (Ours)	0.291 ± 0.024	**50.0**	**41.7**	0.935
No Conflict	**0.295 ± 0.023**	37.5	33.3	**0.944**
No Decay	0.273 ± 0.022	12.5	25.0	0.897
Critic				
PASTA (Ours)	**0.291 ± 0.024**	**75.0**	**45.8**	**0.957**
Branched Unweighted	0.264 ± 0.019	25.0	20.8	0.895
Shared Unweighted	0.246 ± 0.018	0.0	20.8	0.840
Shared Weighted	0.253 ± 0.017	0.0	12.5	0.868
Smoothness				
PASTA (Ours)	**0.291 ± 0.024**	**37.5**	25.0	0.892
$\mu = 0.01$	0.276 ± 0.007	0.0	4.2	0.858
$\mu = 0.1$	0.278 ± 0.024	12.5	8.3	0.858
$\mu = 0.5$	0.280 ± 0.029	0.0	8.3	0.861
$\mu = 1.0$	0.281 ± 0.031	12.5	16.7	0.869
$\mu = 5.0$	0.290 ± 0.026	37.5	**29.2**	**0.893**
$\mu = 10.0$	0.262 ± 0.025	0.0	8.3	0.826

TABLE III: Sensitivity Analysis Results. Statistics computed over eight preference vectors across five seeds.

Method	Hypervolume ↑	Win Rate ↑ [%]	Obj. Dom. Rate ↑ [%]	DMP AUC ↑
Conflict Ratio κ				
$\kappa = 0.4$ (Ours)	**0.291 ± 0.024**	**50.0**	**37.5**	**0.929**
$\kappa = 0.2$	0.266 ± 0.020	0.0	20.8	0.874
$\kappa = 0.6$	0.276 ± 0.019	37.5	16.7	0.901
$\kappa = 0.8$	0.280 ± 0.016	12.5	25.0	0.907
Moving Average λ				
$\lambda = 0.05$ (Ours)	**0.291 ± 0.024**	**62.5**	33.3	**0.941**
$\lambda = 0.2$	0.265 ± 0.026	25.0	29.2	0.881
$\lambda = 0.5$	0.281 ± 0.014	12.5	**37.5**	0.918
Maintenance Rate ρ				
$\rho = 0.15$ (Ours)	**0.291 ± 0.024**	25.0	**29.2**	0.898
$\rho = 0.05$	0.261 ± 0.025	0.0	4.2	0.831
$\rho = 0.3$	0.276 ± 0.024	12.5	16.7	0.860
$\rho = 0.5$	0.281 ± 0.020	25.0	20.8	0.864
$\rho = 0.8$	0.291 ± 0.024	**37.5**	29.2	**0.900**

D. Evaluation on MO-Gym Environments

To evaluate our method in additional settings, we pick continuous high-dimensional environments that have been used to assess on-policy algorithms' performance in widely used libraries like Stable Baselines 3 [38]. Specifically, we use the multi-objective instantiations of MuJoCo's Half Cheetah, Humanoid, Ant and Walker 2D, available through MO-Gym

[12] and shown in Fig. 4. To ensure a fair comparison, we pick as the MOP preferences the same as the original environments, normalized between 0 and 1. Fig. 3 shows the hypervolume comparison for our method and the baselines in the four environments. Overall, we can see PASTA achieves a better performance. See Appendix J for more details and results.

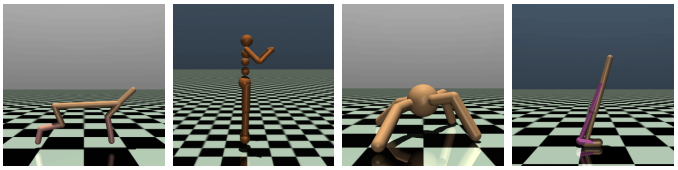

Fig. 4: Multi-objective MuJoCo evaluation environments.

E. Quadrotor Flights among Random Agents

To demonstrate the real-world applicability of PASTA across different platforms, we deployed it on a quadrotor testbed, shown in Fig. 5 and **Video 2**[2]. Specifically, we evaluate our approach in two multi-objective Crazyflie environments: *Frogger* and *Formation*.

In the *Frogger* scenario, the ego drone must traverse a shared workspace while avoiding two other drones executing independent tasks within assigned lanes. Relying solely on a history of the other agents' positions, the ego agent's policy must predict their behavior and plan accordingly to safely weave through active traffic composed of decentralized, independent actors. In the *Formation* scenario, a team of 3 Crazyflies must navigate from a starting zone to a target zone. Throughout the maneuver, the team must maintain an equilateral triangle formation, avoid a dynamic obstacle, and strictly remain within the bounds of the flight arena.

A central challenge in both environments is the presence of dynamic opponents acting as stochastic obstacles. These opponents are governed by a 1D patrolling policy and are constrained to move along a fixed horizontal axis (e.g., $y \in \{-0.3, 0.3\}$ for the *Frogger* task, and $y = 0$ for the *Formation* task) with a constant speed magnitude ($|\dot{x}| \in \{0.03, 0.04\}$ m/s). Their motion follows a "bounce" logic, where velocity is deterministically inverted upon reaching the arena boundaries ($x_{\text{lim}} \approx \pm 0.95$). Additionally, to simulate unpredictable real-world disturbances, there is a 5% probability at every timestep that an opponent will spontaneously reverse its direction. This combination of boundary-constrained oscillation and stochastic direction switching requires the agents to learn collision avoidance policies capable of adapting to sudden movements.

Fig. 6 illustrates the HV achieved across both tasks. Overall, our method yielded higher HV compared to the baselines. Further details regarding quadrotor deployment, RL environments, and comprehensive results are provided in Appendix K.

VII. Discussion and Conclusion

In this work, we addressed the tension between optimization stability and geometric precision in non-convex MORL. We proposed an *Adaptive Smooth Tchebycheff* framework

[2]**Video 2**: https://youtu.be/twh3YZ6eczo

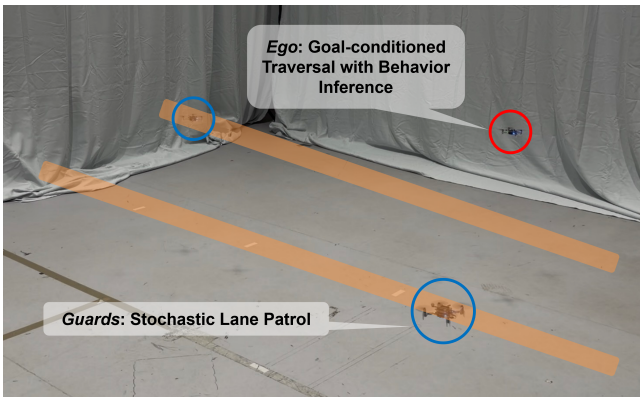

Fig. 5: Goal-conditioned Quadrotor Traversal among Dynamic Obstacles.

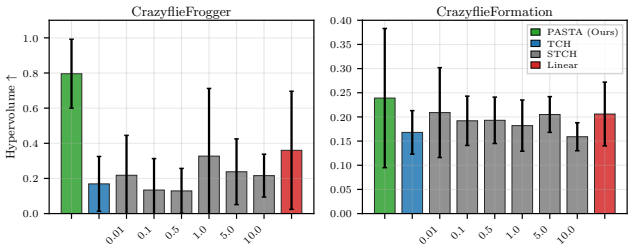

Fig. 6: Performance on Crazyflie Frogger and Formation multi-objective environments. Results are aggregated across five seeds.

that utilizes a novel conflict-driven controller to dynamically modulate the curvature of the scalarization function. By treating gradient interference as a feedback signal, our method anneals towards precise non-convex geometries when feasible, while elastically braking into stable convex approximations upon conflict emergence. Moreover, we introduce the PASTA (Policy-optimization via Adaptive Smooth Tchebycheff Attention) algorithm which integrates the adaptive scalarization into a multi-objective policy's loss by deriving attention weights directly from the STCH analytical derivative. These weights prioritize the value loss toward those with the largest weighted deviation from the utopia point, but also feature a maintenance mechanism to prevent gradient starvation in non-bottleneck objectives. Overall, we showed that our framework provides a robust way for robots to autonomously resolve complex, conflicting requirements in unstructured real-world scenarios, across ground and aerial platforms, as well as in widely used benchmark environments.

Limitations: While our framework enhances stability and geometric precision, it remains dependent on the interplay between the conflict and maintenance rate hyperparameters. The former bottlenecks gradient flow via the smoothness controller, while the latter counteracts this to preserve gradients across all objectives via the attention. The interdependence of these parameters suggests that future work should explore unifying them into a single-parameter or parameter-free feedback mechanism. Additionally, we focused on preference-specific learning; extending to preference-conditioned policies would better support deployment in robotic use cases where priorities could evolve at decision-time.

References

[1] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International conference on machine learning*, pages 11–20. PMLR, 2019.

[2] Joshua Achiam. Spinning Up in Deep Reinforcement Learning, 2018. URL https://spinningup.openai.com/en/latest/.

[3] V Joseph Bowman Jr. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple Criteria Decision Making: Proceedings of a Conference Jouy-en-Josas, France May 21–23, 1975*, pages 76–86. Springer, 1976.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] Andrea Carron, Elena Arcari, Martin Wermelinger, Lukas Hewing, Marco Hutter, and Melanie N Zeilinger. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Robotics and Automation Letters*, 4(4):3758–3765, 2019.

[6] Diqi Chen, Yizhou Wang, and Wen Gao. Combining a gradient-based method and an evolution strategy for multi-objective reinforcement learning. *Applied Intelligence*, 50(10):3301–3317, 2020.

[7] Indraneel Das and John E Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[8] Jean-Antoine Désidéri. Mutiple-gradient descent algorithm for multiobjective optimization. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, 2012.

[9] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[10] Jérémie Dubois-Lacoste, Manuel López-Ibáñez, and Thomas Stützle. Improving the anytime behavior of two-phase local search. *Annals of mathematics and artificial intelligence*, 61(2):125–154, 2011.

[11] Matthias Ehrgott. *Multicriteria optimization*. Springer, 2005.

[12] Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

[13] Florian Felten, El-Ghazali Talbi, and Grégoire Danoy. Multi-objective reinforcement learning based on decomposition: A taxonomy and framework. *Journal of Artificial Intelligence Research*, 79:679–723, 2024.

[14] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51(3):479–494, 2000.

[15] Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys (CSUR)*, 54(6):1–42, 2021.

[16] Nyoman Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242, 2018.

[17] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *arXiv preprint arXiv:2103.09568*, 2021.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Thomas M Howard and Alonzo Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, 26(2): 141–166, 2007.

[20] Xiaoliang Hu, Pengcheng Guo, Yadong Li, Guangyu Li, Zhen Cui, and Jian Yang. Tvdo: Tchebycheff value-decomposition optimization for multiagent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[21] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[22] Sinan Ibrahim, Mostafa Mostafa, Ali Jnadi, Hadi Salloum, and Pavel Osinenko. Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications. *IEEE Access*, 2024.

[23] Avik Jain, Lawrence Chan, Daniel S Brown, and Anca D Dragan. Optimal cost design for model predictive control. In *Learning for Dynamics and Control*, pages 1205–1217. PMLR, 2021.

[24] Hassan Jardali, Durgakant Pushp, Youwei Yu, Mahmoud Ali, Ihab S Mohamed, Alejandro Murillo-Gonzalez, Paul D Coen, Md Al-Masrur Khan, Reddy Charan Pulivendula, Saeoul Park, et al. From zero to high-speed racing: An autonomous racing stack. *arXiv preprint arXiv:2512.06892*, 2025.

[25] Se Hwan Jeon, Steve Heim, Charles Khazoom, and Sangbae Kim. Benchmarking potential based rewards for learning humanoid locomotion. *arXiv preprint arXiv:2307.10142*, 2023.

[26] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[27] Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. *arXiv preprint arXiv:2402.19078*, 2024.

[28] Xi Lin, Yilu Liu, Xiaoyuan Zhang, Fei Liu, Zhenkun

Wang, and Qingfu Zhang. Few for many: Tchebycheff set scalarization for many-objective optimization. *International Conference on Learning Representations (ICLR)*, 2025.

[29] Ni Mu, Yao Luan, and Qing-Shan Jia. Preference-based multi-objective reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 2025.

[30] Alejandro Murillo-González, Junhong Xu, and Lantao Liu. Learning causal structure distributions for robust planning. *IEEE Robotics and Automation Letters*, 2025.

[31] Alejandro Murillo-González and Lantao Liu. Action Flow Matching for Continual Robot Learning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.026.

[32] Alejandro Murillo-González and Lantao Liu. Situationally-aware dynamics learning. *The International Journal of Robotics Research*, 0(0):02783649261431863, 2026. doi: 10.1177/02783649261431863. URL https://doi.org/10.1177/02783649261431863.

[33] VD Noghin. Linear scalarization in multi-criterion optimization. *Scientific and Technical Information Processing*, 42(6):463–469, 2015.

[34] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *arXiv preprint*, 2019.

[35] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

[36] James A. Preiss*, Wolfgang Hönig*, Gaurav S. Sukhatme, and Nora Ayanian. Crazyswarm: A large nano-quadcopter swarm. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304. IEEE, 2017. doi: 10.1109/ICRA.2017.7989376. URL https://doi.org/10.1109/ICRA.2017.7989376. Software available at https://github.com/USC-ACTLab/crazyswarm.

[37] Shuang Qiu, Dake Zhang, Rui Yang, Boxiang Lyu, and Tong Zhang. Traversing pareto optimal policies: Provably efficient multi-objective reinforcement learning. *arXiv preprint arXiv:2407.17466*, 2024.

[38] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[39] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto conditioned networks. *arXiv preprint arXiv:2204.05036*, 2022.

[40] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[41] Stefan Schäffler, Reinhart Schultz, and Klaus Weinzierl. Stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114(1):209–222, 2002.

[42] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[43] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[44] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[45] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[46] Jonaid Shianifar, Michael Schukat, and Karl Mason. Adaptive scalarization in multi-objective reinforcement learning for enhanced robotic arm control. *Neurocomputing*, page 132205, 2025.

[47] Ralph E Steuer and Eng-Ung Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26(3):326–344, 1983.

[48] Ralph E Steuer and Eng-Ung Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, 1983.

[49] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[50] Huan Tran Thien, Cao Van Kien, and Ho Pham Huy Anh. Optimized stable gait planning of biped robot using multi-objective evolutionary jaya algorithm. *International Journal of Advanced Robotic Systems*, 17(6):1729881420976344, 2020.

[51] Bart van Marum, Aayam Shrestha, Helei Duan, Pranay Dugar, Jeremy Dao, and Alan Fern. Revisiting reward design and evaluation for robust humanoid standing and walking. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11256–11263. IEEE, 2024.

[52] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

[53] Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pages 191–199. IEEE, 2013.

[54] Kristof Van Moffaert, Tim Brys, Arjun Chandra, Lukas Esterle, Peter R Lewis, and Ann Nowé. A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 2306–2314. IEEE, 2014.

[55] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)*, 8 (2):1029–1036, 2023. doi: 10.1109/LRA.2023.3236571.

[56] Nils Wilde, Stephen L Smith, and Javier Alonso-Mora. Scalarizing multi-objective robot planning problems using weighted maximization. *IEEE Robotics and Automation Letters*, 9(3):2503–2510, 2024.

[57] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

[58] Jiangjiao Xu, Ke Li, and Mohammad Abusara. Preference based multi-objective reinforcement learning for multi-microgrid system optimization problem in smart grid. *Memetic Computing*, 14(2):225–235, 2022.

[59] Jingyun Yang, Max Sobol Mark, Brandon Vu, Archit Sharma, Jeannette Bohg, and Chelsea Finn. Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4804–4811. IEEE, 2024.

[60] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

[61] Yucheng Yang, Tianyi Zhou, Mykola Pechenizkiy, and Meng Fang. Preference controllable reinforcement learning with advanced multi-objective optimization. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=49g4c8MWHy.

[62] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[63] Youwei Yu, Junhong Xu, and Lantao Liu. Adaptive diffusion terrain generator for autonomous uneven terrain navigation. *arXiv preprint arXiv:2410.10766*, 2024.