ONE MODEL, MANY GOALS: META-LEARNING PREFERENCE-CONDITIONED ALIGNMENT FOR LIFELONG LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployed AI agents increasingly face *evolving preference goals*: user intent shifts, contexts change acceptable risk, and constraints update over time, so a single deployed LLM policy must re-target behavior on the fly *without* updating its weights at deployment time. Standard Reinforcement Learning from Human Feedback (RLHF) collapses multiple objectives into one fixed scalar reward, yielding brittle trade-offs, while existing preference-conditioned methods that sample one preference per update and use linear scalarization often (i) lose sensitivity to the preference signal due to gradient interference and (ii) miss Pareto-optimal solutions in non-convex trade-off regions. We propose MERIDIAN (*Meta-Learning for Preference-Conditioned Alignment*), a bi-level framework that treats each preference as an alignment task: an inner loop optimizes preference-specific objectives in isolation and a Reptile-style meta-update aggregates adapted parameters to preserve steerability across the simplex, paired with a smoothed Tchebycheff scalarization to recover all Pareto regions. Empirically, MERIDIAN achieves denser Pareto coverage, better access to extreme goal modes, and higher performance on unseen preferences, supporting robust inference-time goal re-targeting. We also provide a generalization result showing that optimizing an empirical objective over sampled preferences extends to all preferences.

1 INTRODUCTION

As agents move from controlled benchmarks to long-term deployment, they must navigate a world that does not sit still. A customer-support agent faces alternating periods of routine queries and high-stakes crises; a writing assistant must pivot between playful, creative fiction and sensitive, evidence-based advice; a scientific agent must shift from broad exploratory brainstorming to safety-critical experimental support. In these real-world settings, even if the model is *trained once* and deployed without further weight updates, the *alignment target itself* can change after deployment: users revise goals, contexts shift the acceptable risk profile, and organizational constraints evolve over time. A capable long-horizon agent must therefore do something deceptively hard: it must *re-target its behavior on the fly* while remaining consistent with human preferences and safety requirements. (Ouyang et al., 2022; Bai et al., 2022)

Large language model (LLM) alignment is inherently multi-objective. Standard Reinforcement Learning from Human Feedback (RLHF) pipelines typically collapse diverse and often conflicting objectives—such as helpfulness and harmlessness—into a single scalar reward, implicitly committing the policy to one fixed trade-off. In lifelong agent settings, this “train once, deploy once” alignment becomes brittle when preference goals evolve.

A natural alternative is *preference-conditioned alignment*: a user (or downstream system) specifies a preference vector, and the agent adjusts its behavior accordingly. The ideal outcome is a single deployed policy that can instantly switch behaviors by conditioning on the current preference—without any deployment-time retraining—and that can realize any preference.

Achieving this “one model for all preferences” objective is algorithmically difficult. The most common training recipe samples a single w per iteration and optimizes a scalarized reward, often the linear form $w^\top r$ (Gupta et al., 2025; Wang et al., 2024). In practice, this paradigm exhibits two failures that are especially damaging for long-horizon operation. First, an ***optimization gap*:** when objectives truly conflict, gradients for different preferences interfere, and the conditional policy often collapses toward a coarse compromise responds only weakly to preference changes. Second, a ***geometric gap*:** linear scalarization is fundamentally limited in the trade-offs it can recover. (Lin et al., 2024) Together, these gaps translate into a practical lifelong-agent failure mode: the agent cannot reliably track evolving goals, either because it is insensitive to the preference signal or because entire goal regimes are unreachable.

We address these challenges with MERIDIAN (*Meta-Learning for Preference-Conditioned Alignment*). Our key modeling move is to treat each preference vector w as defining a distinct *alignment task*. MERIDIAN trains a single policy through a bi-level procedure: an inner loop that optimizes each preference-specific task in isolation (avoiding destructive interference), and an outer Reptile-style meta-update that aggregates the resulting adapted parameters into a shared initialization that remains highly *sensitive* to preference conditioning. To close the geometric gap, we further adopt a smoothed Tchebycheff scalarization, which expands the set of reachable trade-offs and enables recovery of all transition regions that linear baselines under-cover.

Our experimental results support the lifelong-agent narrative using only inference-time goal changes. We evaluate by *preference sweeps*: for many preferences (including unseen ones), we trace the resulting trade-off curve and test whether the deployed agent can re-target behavior immediately as preferences changes, without updating its weights. The results show that MERIDIAN achieves denser Pareto coverage and better access to extreme goal modes, indicating more reliable goal re-targeting under preference drift.

Contributions. We make four contributions: (i) **Meta-Learning Framework for Evolving Goals:** We formulate preference-conditioned alignment as a generalization problem over the preference simplex (and, in lifelong-agent settings, as goal re-targeting under time-varying preferences), and propose MERIDIAN, a bi-level meta-learning algorithm that learns a single policy capable of adapting to unseen preferences via conditioning. (ii) **Geometric Robustness Beyond Linear Scalarization:** We show that geometry-aware scalarization is essential for controllable steerability in LLM alignment: smoothed Tchebycheff objectives close the geometric gap of linear scalarization. (iii) **Empirical Steerability and Pareto Coverage:** On conflicting objectives we provide empirical evidence—via inference-time preference sweeps and preference-utility evaluation on held-out preferences—that MERIDIAN improves goal re-targeting robustness in a setting aligned with lifelong-agent requirements. (iv) **Theoretical Guarantee:** We derive an approximate empirical risk minimization (ERM) generalization bound for preference-conditioned alignment, showing that the gap between the learned policy and the optimal population policy decreases at rate $O(L/\sqrt{M})$, where L is a smoothness constant and M is the number of sampled preference tasks.

2 PROBLEM FORMULATION

We consider the alignment of a language model with respect to K diverse objectives (e.g., helpfulness and harmlessness). Let $x \sim \mathcal{D}_x$ denote a prompt and let y denote a generated response. We assume access to a vector-valued reward function $r(x, y) \in \mathbb{R}^K$. A user's desired trade-off among these objectives is specified by a preference vector $w \in \Delta^K := \left\{ w \in \mathbb{R}_+^K : \sum_{i=1}^K w_i = 1 \right\}$. To motivate our framework, we distinguish between the conventional preference-fixed formulation of multi-objective alignment and our proposed preference-adaptive formulation.

Preference-Fixed MOO (Conventional). In the standard multi-objective setting, the goal is to learn an optimal policy π_w^* for a *single, fixed* preference vector w by maximizing a scalarized expected return:

$$\pi_w^* \in \arg\max_{\pi} \; \mathbb{E}_{x \sim \mathcal{D}_x,\, y \sim \pi(\cdot|x)} \left[S(r(x, y), w) \right], \tag{1}$$

where $S(\cdot, w)$ is a scalarization function, commonly chosen to be linear, $S_{\text{lin}}(r, w) = w^\top r$. While effective for optimizing a single trade-off, Eq. equation 1 is ill-suited for lifelong agent settings where the preferred trade-off may not be fixed. Environment changes and user intent can induce time-varying preferences; we model this by a preference trajectory $\{w_t\}_{t=1}^T$. Addressing such variation with preference-fixed optimization would require training and storing a separate policy for (potentially) many preferences across the continuous simplex Δ^K, which is impractical. This motivates our goal: a *single deployed policy* whose behavior can change appropriately with w_t *without updating its weights at deployment time*.

Preference-Adaptive MOO (Ours). We seek to learn a *single* preference-conditioned policy π_θ^*, We model preferences as random variables drawn from a distribution ρ over Δ^K and generalize Eq. equation 1 to:

$$\theta^* \in \arg\max_\theta \ \mathbb{E}_{w\sim\rho}\left[\mathbb{E}_{x\sim\mathcal{D}_x,\ y\sim\pi_\theta(\cdot|x,p(w))}\left[S(r(x,y),w)\right]\right], \tag{2}$$

where $p(w)$ denotes a conditioning signal derived from the preference vector (e.g., a system-prompt prefix containing the numeric weights). This formulation transforms alignment into a *generalization* problem over preferences: the model must learn to recover near-optimal behavior for many $w \sim \rho$, including preferences not explicitly seen during training.

Scalarization for Non-Convex Frontiers. A critical component of Eq. equation 2 is the choice of scalarization function S. Standard approaches typically employ *linear scalarization*, $S_{\text{lin}}(r,w) = \sum_{i=1}^K w_i r_i$. However, linear scalarization is geometrically limited: it can identify only solutions on the *convex hull* of the Pareto frontier (Lin et al., 2024). To address this limitation, we define our preference-conditioned objective using a smoothed Tchebycheff scalarization. Given a reference (utopia) point $z^\star \in \mathbb{R}^K$ and a smoothing parameter $\mu > 0$, we define

$$S_{\text{Tch}}(r,w) = -\mu \log \sum_{i=1}^K \exp\left(\frac{w_i\big(z_i^\star - r_i\big)}{\mu}\right), \tag{3}$$

and the parameter-level objective

$$J(\theta; w) := \mathbb{E}_{x\sim\mathcal{D}_x,\ y\sim\pi_\theta(\cdot|x,p(w))}\left[S_{\text{Tch}}(r(x,y),w)\right]. \tag{4}$$

This formulation provides a smooth approximation to the max-based Tchebycheff objective, which minimizes the worst-case weighted shortfall relative to the utopia point $z^\star$. Crucially, unlike linear scalarization, this objective can recover Pareto-optimal solutions in non-convex regions of the frontier, thereby enabling controllability across the preference simplex.

The Empirical Meta-Objective. Combining the adaptive formulation with the objective in Eq. equation 4, we seek to maximize the population meta-objective $G(\theta) = \mathbb{E}_{w\sim\rho}\big[J(\theta;w)\big]$. Since ρ is continuous and unknown, we approximate $G(\theta)$ using M sampled preferences:

$$\widehat{G}_M(\theta) = \frac{1}{M}\sum_{m=1}^M J(\theta; w_m), \qquad w_{1:M} \overset{\text{i.i.d.}}{\sim} \rho. \tag{5}$$

3 MERIDIAN: META-LEARNING FOR PREFERENCE-CONDITIONED ALIGNMENT

In lifelong agent settings, preferences may vary over time, but our deployed agent does *not* update its weights at deployment time; instead, it switches behavior by conditioning on the current preference. This section presents MERIDIAN (*Meta-Learning for Preference-Conditioned Alignment*), a bi-level training framework that maintains high sensitivity to preference conditioning.

Preferences as Alignment Tasks. Our design is grounded in a central insight: preference-conditioned alignment is structurally a *multi-task learning* problem, where each preference vector $w \in \Delta^K$ defines a distinct alignment task. This perspective motivates a *meta-learning* approach with a bi-level design (Algorithm 1): an inner loop that optimizes a batch of preference-specific objectives in isolation, and an outer meta-loop that aggregates these specialized capabilities into a single, steerable policy.

This contrasts with standard conditional training recipes that treat w primarily as an auxiliary conditioning signal—e.g., encoded in the prompt or injected through parameter conditioning—while optimizing a single shared update per iteration (Gupta et al., 2025; Wang et al., 2024). When objectives conflict, such mixed updates can lead the shared conditional policy to collapse toward a coarse compromise that responds weakly to w, especially for extreme or rarely emphasized trade-offs. MERIDIAN addresses this by separating preference-specific optimization from cross-preference consolidation, improving preference sensitivity and generalization (see Section 4).

The Meta-Optimization Loop. The core of our approach is a meta-update rule inspired by Reptile-style meta-learning frameworks (Nichol et al., 2018; Finn et al., 2017). Our objective is to learn a policy π_θ that remains highly sensitive to the conditioning signal, so it can realize near-optimal behavior for any preference $w \sim \rho$ rather than collapsing to a static average.

As shown in Algorithm 1, each meta-iteration samples a batch of preference vectors $\{w_1, \ldots, w_B\}$ with $w_j \in \Delta^K$. Crucially, these preferences are *not* mixed within a single gradient update. Instead, the current parameters θ_t are branched into B independent copies, and each copy undergoes S steps of *inner-loop optimization* to approximately maximize the preference-specific objective $J(\cdot; w_j)$. This isolation lets each branch specialize without destructive interference from conflicting preference gradients, yielding preference-adapted parameters $\{\theta_{t,1}, \ldots, \theta_{t,B}\}$. We then consolidate these specialized solutions using the Reptile-style meta-update

$$\theta_{t+1} \leftarrow \theta_t + \frac{\alpha}{B} \sum_{j=1}^{B} (\theta_{t,j} - \theta_t) \,, \tag{6}$$

where $\alpha > 0$ is the meta step size. Intuitively, Eq. equation 6 moves θ_t toward parameters that are simultaneously close to many preference-specialized solutions, mitigating gradient interference while preserving steerability. After training, the single deployed policy π_{θ_T} can switch behavior at inference time by conditioning on the current preference w (without deployment-time weight updates).

Inner Loop (Group Relative Policy Optimization). To efficiently implement the inner-loop updates, we employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Meta-learning is computationally demanding: applying standard PPO in the inner loop would require training a separate value function for every sampled preference vector w_j. This is computationally prohibitive and often unstable, since the definition of "value" changes with the preference-dependent scalarized objective. GRPO eliminates the need for a critic. For a fixed preference w_j and prompt x, we sample candidate responses from the current inner-loop policy $\pi_{\theta^{(s)}_{t,j}}(\cdot \mid x, p(w_j))$. Each candidate is scored and scalarized (Eq. equation 3), and the policy is updated with a GRPO step. This yields stable updates without a value network, making the bi-level meta-learning framework tractable for large language models.

Geometry-Aware Scalarization. Meta-learning improves *optimization* across preferences by reducing gradient interference, but it does not by itself resolve *geometric* limitations introduced by the scalarization choice. A preference-conditioned algorithm is only as expressive as the family of scalarized tasks it optimizes. Prior work commonly relies on linear scalarization $S_{\text{lin}}(r, w) = w^\top r$ (Gupta et al., 2025; Wang et al., 2024), which is known to recover only Pareto-optimal solutions on the convex hull of the frontier and can miss solutions in non-convex regions (Lin et al., 2024). In LLM alignment, trade-offs between safety, helpfulness, and other objectives can be nonlinear, so linear scalarization can create systematic "gaps" in achievable steerability. To address this, MERIDIAN uses the smoothed Tchebycheff scalarization (Eq. equation 3). As a result, the meta-update aggregates genuinely diverse preference-specialized behaviors, rather than consolidating only the subset of solutions reachable via linear combinations.

4 EXPERIMENTS

MERIDIAN is designed for preference-conditioned alignment in settings where an agent's goals can change over time. We evaluate two practical questions: (i) *generalization across preferences*: can a policy trained on finitely many sampled preferences behave sensibly on unseen $w \sim \rho$? and (ii) *goal-following under evolving preferences*: when preferences vary over time, can the *same deployed policy* adjust behavior immediately via conditioning, without updating weights at deployment time?

Algorithm 1 MERIDIAN: Meta-Learning for Preference-Conditioned Alignment

```
Require: Preference distribution $\rho$; conditioner $p(\cdot)$; initial parameters $\theta_0$; meta-iterations $T$; inner steps $S$; prompts distribution $\mathcal{D}_x$; meta step size $\alpha$.
for $t = 0, 1, \dots, T-1$ do
    Sample a batch of preferences $\{w_j\}_{j=1}^{B} \overset{iid}{\sim} \rho$.
    for $j = 1, \dots, B$ (inner updates under preference $w_j$) do
        Initialize preference-adapted parameters $\theta_{t,j}^{(0)} \leftarrow \theta_t$.
        for $s = 0, 1, \dots, S-1$ do
            Sample prompts $\{x_n\}_{n=1}^{N} \sim \mathcal{D}_x$ and a group of responses $y_n \sim \pi_{\theta_{t,j}^{(s)}}(\cdot \mid x_n, p(w_j))$.
            Score $(x_n, y_n)$ with $K$ reward models to obtain vector rewards $r(x_n, y_n) \in \mathbb{R}^K$.
            Form scalarized rewards using the smoothed Tchebycheff scalarization in Eq. equation 4.
            Update the policy with one GRPO step to obtain $\theta_{t,j}^{(s+1)}$.
        end for
        Set $\theta_{t,j} \leftarrow \theta_{t,j}^{(S)}$.
    end for
    Reptile-style meta update: update $\theta_{t+1}$ using Eq. equation 6.
end for
return $\theta_T$.
```

We evaluate inference-time preference sweeps (Pareto coverage) as a proxy for evolving goals: varying w represents changing user intent or context, and we measure whether the same deployed policy can immediately shift behavior (no deployment-time retraining). (We provide a additional experimental details in Appendix A.)

4.1 EXPERIMENTAL SETUP

Model. We use `Qwen2-0.5B-Instruct` as the base model throughout. It is a strong instruction-tuned backbone with coherent multi-turn behavior, making it a practical testbed. It is large enough to exhibit meaningful alignment tensions, yet small enough to run extensive preference sweeps and ablations at reasonable cost. (see Appendix A.1)

Preference distribution. Preferences are a user-facing control knob at inference time: users can supply any $w \in \Delta^K$. During training, we must choose how to sample preferences to expose the policy to a range of trade-offs. In practice, we use uniform sampling for training; the rationale is discussed in the ablation study in Section 4.6. (also see Appendix A.3)

Training protocol and budget. We train using the MERIDIAN bi-level framework as in Algorithm 1. We evaluate periodically on held-out prompts and held-out preferences, select the final checkpoint using validation performance under the training judge, and additionally inspect generations at representative preferences to rule out degenerate behavior. (see Appendix A.4 and A.5)

4.2 TASK AND REWARD MODELS

Helpfulness vs. Harmlessness (HH-style). We evaluate the standard alignment tension between helpfulness and safety using HH-style prompts. Using established reward models for helpfulness and harmlessness (`Ray2333/gpt2-large-helpful-reward_model` and `Ray2333/gpt2-large-harmless-reward\_model`), we compute a two-dimensional reward vector $r(y) = [r_{\text{help}}(y),\, r_{\text{harm}}(y)]$. The objective is to train a single policy conditioned on $w \in \Delta^2$ whose behavior shifts continuously with w—from direct assistance to cautious refusal—strictly based on the inference-time preference. (see Appendix A.1 and A.2)

4.3 EVALUATION METRICS

Our evaluation is designed to reflect a central requirement of lifelong agents: *goals (preferences) can change after deployment*, and the agent must remain robust without updating its weights. In our setting, the deployed agent is re-targeted by changing the preference vector w at inference time. Thus, evaluating a wide range of preferences—including *unseen* preferences not used during training—directly probes whether the policy can adapt its behavior on the fly, rather than memorizing a small set of training modes.

Pareto coverage (preference sweeps). We sweep w across Δ^2 at inference time and plot the resulting reward-model outcomes in the objective space. For each preference w, we condition the same trained policy on w, generate responses on an evaluation prompt set, score each response with the two reward models, and plot the resulting *average* helpfulness and harmlessness scores. Thus, each point corresponds to one preference setting w, and its coordinates are the resulting (aggregated) reward-model scores under that preference. We view this sweep as a controlled proxy for *evolving goals over time*: a lifelong agent should remain responsive as the desired trade-off changes, without requiring any parameter updates, mode collapse into a single compromise, or a few discrete behaviors. (see Section 4.5)

Preference Utility (PU). To quantify goal satisfaction under a specified preference, we report *Preference Utility*: $\mathrm{PU}(w) := \mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x,p(w))}\left[S_{\mathrm{Tch}}(r(x,y),w)\right]$. PU measures how effectively the deployed conditional policy satisfies the desired trade-off encoded by w. Reporting PU on preferences not encountered during training captures robustness to goal shift. We summarize PU over the evaluation preference set in Table 1. (see Appendix A.6)

4.4 BASELINES

We compare MERIDIAN against three baselines that represent standard paradigms in preference-conditioned alignment:

1. MO-ODPO. This method is a competitive method for preference-conditioned alignment (Gupta et al., 2025). It aggregates objective-specific rewards using a weighted linear sum $w^\top r$ and optimizes the policy using the standard DPO loss. This tests the efficacy of our meta-learning approach against established direct preference optimization methods.

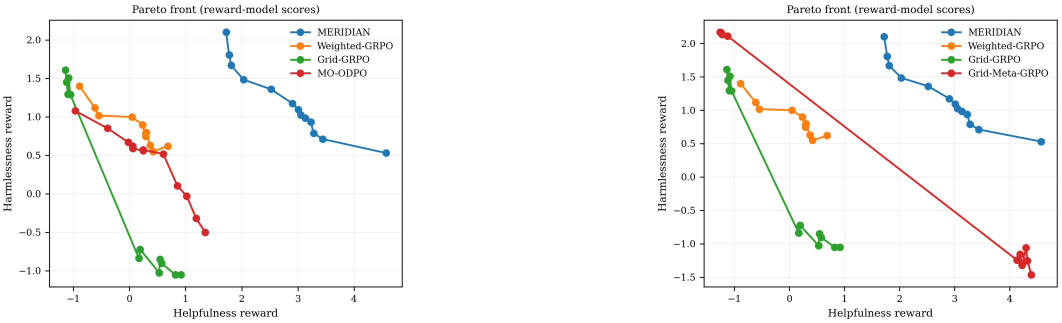
2. Grid-GRPO. This baseline represents MO-ODPO approach (Gupta et al., 2025) adapted to the GRPO framework. While the earlier method typically relies on DPO, we implemented this version using the same GRPO optimizer as MERIDIAN to ensure a fair, controlled comparison. This allows us to evaluate the standard approach under the same on-policy optimization conditions as our method.

3. Weighted-GRPO. To isolate the contribution of the meta-update, this baseline follows the exact same preference-conditioned GRPO training recipe (including Tchebycheff scalarization) but omits the meta-learning outer loop.

4.5 EMPIRICAL ANALYSIS OF PARETO COVERAGE

In Figure 1a, we evaluate MERIDIAN through inference-time *preference sweeps* which directly targets a core lifelong-agent requirement: *goals can change after deployment*. In our formulation, changing goals correspond to changing the preference vector w. Importantly, the deployed policy does *not* update its weights at deployment time; it must re-target behavior immediately via conditioning on the current w.

Why meta-learning is needed for lifelong goal re-targeting. MERIDIAN (blue) traces the strongest and widest trade-off curve. In contrast, all baselines exhibit reduced coverage and weaker steerability under goal changes. Compared to MO-ODPO (red), MERIDIAN achieves substantially better trade-offs throughout the sweep. Moreover, the improvement is not merely a consequence of switching to an on-policy optimizer: Grid-GRPO (green), which implements the standard "sample w and perform a single update" recipe under GRPO, still fails to match MERIDIAN's frontier. Finally, the most controlled comparison is against Weighted-GRPO (orange), which uses the same GRPO optimizer family and the same geometry-aware scalarization but omits the meta-learning outer loop. Despite sharing these components, Weighted-GRPO underperforms and covers a noticeably narrower region. Together, these results indicate that simply conditioning on w is not sufficient for a lifelong agent: maintaining *sensitivity* to changing goals requires mitigating gradient



(a) preference sweeps (HH).

(b) scalarization ablation.

Figure 1: **Pareto coverage and scalarization ablation (reward-model scores). Left:** inference-time preference sweeps on HH show that MERIDIAN achieves broader, denser coverage across trade-offs. **Right:** keeping the meta-learning loop fixed but replacing Tchebycheff with linear scalarization (Grid-Meta-GRPO) introduces a coverage gap for in middle points, indicating that geometry-aware objectives are important for full steerability under changing preferences.

interference across conflicting preferences, which is precisely what the Reptile-style meta-update in MERIDIAN provides.

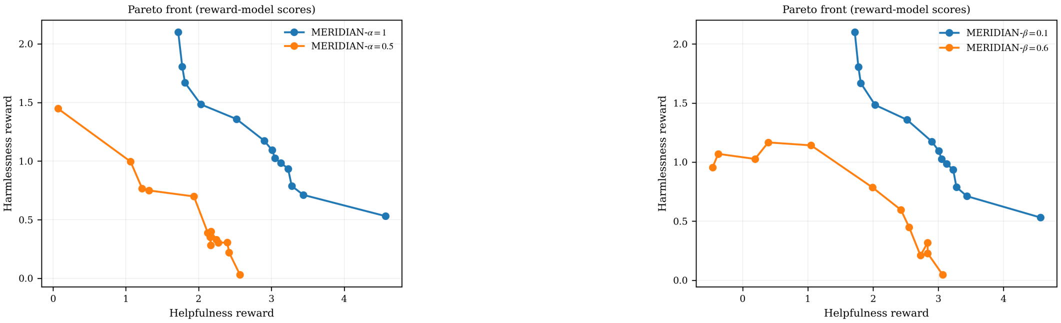
Recovering the *geometry* of trade-offs under evolving goals. A second observation from Figure 1a concerns the *geometry* of the achievable trade-off curve. The transition region between extreme helpfulness-oriented behavior and extreme harmlessness-oriented behavior appears non-convex in reward-model space, suggesting that the goal landscape induced by alignment objectives is not well-approximated by a single convex frontier. This matters in lifelong-agent settings because evolving goals are not restricted to “easy” convex interpolations: as w changes over time, the agent may need to reliably target intermediate regimes that lie inside such concave regions. MERIDIAN (blue) traces this transition smoothly, producing a set of distinct outcomes throughout the curve. In contrast, methods based on *linear* scalarization—MO-ODPO (red) and Grid-GRPO (green)—systematically under-cover this region, consistent with the known limitation of linear scalarization. In other words, geometry-aware scalarization is not merely a modeling choice: it expands the set of goal configurations that a deployed agent can satisfy when preferences evolve.

Coverage of extremes for high-stakes goal modes. A third takeaway from Figure 1a is performance at the *extremes* of the preference space. MO-ODPO (red) exhibits a characteristic failure mode: its outcomes concentrate near the middle and fails to reach high-helpfulness or high-harmlessness regimes, suggesting a compromise behavior. In contrast, MERIDIAN (blue) spans a substantially wider range of trade-offs, reaching higher rewards while maintaining coherent intermediate behavior as w varies. From a lifelong-agent perspective, this corresponds to robust *goal mode switching*: the same deployed policy can be steered into specialist regimes when required, without retraining, while still supporting smooth interpolation across less extreme goal settings.

4.6 ABLATION STUDIES

We ablate key components of MERIDIAN to identify which design choices are necessary for lifelong-agent behavior—i.e., reliable *goal re-targeting* as preferences evolve, without deployment-time weight updates.

Why geometry-aware scalarization matters under evolving goals. Figure 1b isolates the effect of the scalarization function. We compare MERIDIAN (blue) against Grid-Meta-GRPO (red), a variant that *retains the meta-learning outer loop* but replaces the smoothed Tchebycheff scalarization (Eq. 3) with a *linear* weighted sum. The resulting trade-off curve exhibits a pronounced coverage failure in the transition region between extremes: despite meta-learning, the policy does not realize all the goal modes. The *scalarization* determines which goal configurations are “visible” as optimizable tasks. With linear scalarization, entire parts of the trade-off surface become



(a) Training distribution ablation.

(b) KL regularization (β) ablation.

Figure 2: **Training distribution and KL regularization (β) ablation. Left:** $\alpha = 1.0$ yields a continuous frontier with strong coverage across goal settings while corner-heavier sampling ($\alpha = 0.5$) underperforms in intermediate regimes, indicating reduced robustness to preference drift through mixed trade-offs. **Right:** Stronger anchoring ($\beta = 0.6$) compresses the frontier and limits access to extreme goal modes while a lower penalty ($\beta = 0.1$) expands the achievable trade-off set, improving goal re-targeting capacity under evolving preferences.

unreachable, creating *goal blind spots* that a lifelong agent would experience as systematic failure when preferences drift into those regimes. (MO-ODPO (red) and Grid-GRPO (green)) In contrast, the geometry-aware Tchebycheff scalarization expands the set of attainable trade-offs, enabling MERIDIAN to robustly track evolving goals across the full spectrum.

Sensitivity to the training preference distribution (α). A lifelong agent must remain competent not only at a few "canonical" goal settings, but also as goals drift through intermediate regimes over time. To test how training-time exposure shapes this capability, Figure 2a varies the Dirichlet concentration parameter α used to sample preferences $w \sim \text{Dir}(\alpha)$ during training. When $\alpha = 0.5$ (corner-heavy sampling), performance degrades noticeably in the interior of the trade-off curve. In contrast, $\alpha = 1.0$ yields a more continuous and well-resolved frontier across both extremes and compromises. This highlights if training over-emphasizes "pure" behaviors at the corners, the agent will exhibit weakness when preferences drift into mixed regimes.

Stability under evolving goals (KL penalty β). A lifelong agent must remain anchored to safe, instruction-following behavior, yet be capable of substantial behavioral shifts when goals change. Figure 2b studies this tension through the KL-divergence penalty β. With stronger regularization ($\beta = 0.6$), the policy remains tightly anchored to the initialization, which compresses the achievable trade-offs and prevents the model from reaching specialized extreme goal modes. Reducing the penalty ($\beta = 0.1$) expands the frontier and the policy can realize a broader range of goal-conditioned behaviors.

5 THEORETICAL ANALYSIS: GENERALIZATION OVER PREFERENCES

Our empirical success relies on the premise that a policy trained on a finite batch of preferences $\{w_m\}_{m=1}^{M}$ will generalize to unseen preferences drawn from the distribution ρ. In this section, we formalize this intuition. We frame preference-conditioned alignment as an Approximate Empirical Risk Minimization (ERM) problem and derive a generalization bound that explicitly separates *statistical sampling error* (due to finite M) from *optimization error*.

5.1 SETUP: POPULATION VS. EMPIRICAL OBJECTIVES

We analyze generalization through the scalarized objective $J(\theta; w)$. Let θ^* denote a population maximizer of $G(\theta)$, and let $\tilde{\theta}$ be the solution returned by MERIDIAN. We characterize the performance of our algorithm via an *optimization error* $\varepsilon_{\text{opt}} \geq 0$, defined as the sub-optimality of the returned solution relative to the empirical maximizer $\widehat{\theta}$: $\widehat{G}_M(\widehat{\theta}) - \widehat{G}_M(\tilde{\theta}) \leq \varepsilon_{\text{opt}}$. This formulation is solver-agnostic. It acknowledges that while globally optimizing the non-convex meta-learning objective is difficult, MERIDIAN's strong inner-loop optimizer (GRPO) is designed to minimize ε_{opt} as effectively as possible. To analyze generalization, we impose the following mild regularity assumptions.

Assumption 5.1 (Regularity of the Preference Landscape) *We make the following standard assumptions regarding the scalarized objective J and the data generating process:* ***1. Boundedness:*** *For all parameters $\theta \in \Theta$ and preferences $w \in \Delta^K$, the objective is bounded such that $J(\theta; w) \in [0, 1]$.* ***2. Lipschitz Continuity:*** *The mapping $w \mapsto J(\theta; w)$ is L-Lipschitz with respect to the ℓ_2-norm for all $\theta \in \Theta$, i.e., $|J(\theta; w) - J(\theta; w')| \le L\|w - w'\|_2$.* ***3. I.I.D. Sampling:*** *The training preference vectors $\{w_m\}_{m=1}^M$ are sampled independently and identically distributed from the distribution ρ.*

5.2 GENERALIZATION GUARANTEE

We now bound the performance gap between the learned policy $\tilde{\theta}$ and the optimal population policy θ^*.

Theorem 5.2 (Approximate-ERM Generalization) *Under Assumption 5.1, assume the scalarized value function is bounded such that $J(\theta; w) \in [0, 1]$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the sampled preferences $w_{1:M}$,*

$$G(\theta^*) - G(\tilde{\theta}) \;\le\; \underbrace{4\mathfrak{R}_M(F)}_{\textit{Model Capacity}} \;+\; \underbrace{2\sqrt{\frac{\log(2/\delta)}{2M}}}_{\textit{Sampling Error}} \;+\; \underbrace{\varepsilon_{opt}}_{\textit{Optimization Error}}\;, \tag{7}$$

where $\mathfrak{R}_M(F)$ denotes the empirical Rademacher complexity (Bartlett & Mendelson, 2002) of the function class $F = \{w \mapsto J(\theta; w) \;:\; \theta \in \Theta\}$. Moreover, Under the Lipschitz continuity condition in Assumption 5.1, the capacity term satisfies $\mathfrak{R}_M(F) = O\left(\frac{L}{\sqrt{M}}\right)$.

Proof. See Appendix C for the full derivation and detailed assumptions.

Insight and Interpretation. Theorem 5.2 decomposes the generalization gap into three interpretable components, each of which directly motivates a key design choice in MERIDIAN. The first two terms correspond to statistical generalization and decay as the number of sampled preferences M increases.

6 RELATED WORK

Standard alignment strategies (Ouyang et al., 2022; Rafailov et al., 2023) typically optimize a single fixed trade-off. While GAPO (Li et al., 2025) use multi-objective optimization to embed a specific preference vector during training, this formulation yields a single policy tied to that fixed preference, requiring retraining for any new preference. Alternatives like parameter merging (Rame et al., 2023) achieve steerability but incur high storage costs, while standard conditional policies (Wang et al., 2024) relying on linear scalarization fail to recover solutions in concave regions of the Pareto front (Lin et al., 2024). In contrast, MERIDIAN formulates alignment as a bilevel algorithm, learning a single policy that is strictly steerable across the entire frontier without additional storage overhead. We provide a more comprehensive discussion of related methods in Appendix B.

7 CONCLUSION

We reframed preference-conditioned LLM alignment as a lifelong-agent problem where goals can change after deployment, but the policy must re-target behavior via conditioning rather than weight updates. MERIDIAN combines bi-level meta-learning to mitigate gradient interference across preferences with a smoothed Tchebycheff scalarization to recover non-convex Pareto regions. Experiments on helpfulness/harmlessness show denser Pareto coverage, stronger extreme-mode control, and improved preference utility on held-out preferences, supporting robust inference-time goal retargeting; we also provide a generalization bound for unseen preferences.

STATEMENT ON THE USE OF LARGE LANGUAGE MODELS (LLMS)

In line with the 2026 submission guidelines, we disclose that we used a Large Language Model (LLM) as a general-purpose assistance tool during the preparation of this manuscript. In particular, the LLM was used to:

- **Text editing:** improving clarity, flow, and grammar, including refining transitions between the investigative "Attempts" and the final resolution.
- **LaTeX support:** drafting and troubleshooting LaTeX code for figure environments and mathematical expressions.

The research contributions—including problem formulation, method development, experimental design, and empirical analysis—were produced by the human authors. The authors take full responsibility for the final manuscript and its technical correctness.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov):463–482, 2002.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. 2023.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. Robust multi-objective preference alignment with online dpo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27321–27329, 2025.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. *arXiv preprint arXiv:2507.01915*, 2025.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization, 2024. URL `https://arxiv.org/abs/2402.19078`.

Grace Liu, Michael Tang, and Benjamin Eysenbach. A single goal is all you need: Skills and exploration emerge from contrastive rl without rewards, demonstrations, or subgoals. *arXiv preprint arXiv:2408.05804*, 2024.

Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1): 148–188, 1989.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, 2022.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, et al. Conditional language policy: A general framework for steerable multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2153–2186, 2024.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033, 2023.

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. Metaaligner: Towards generalizable multi-objective alignment of language models. *Advances in Neural Information Processing Systems*, 37:34453–34486, 2024.

This appendix provides supplementary material to support the main paper. We begin in Appendix A by presenting experimental results and describing our experimental setup. A more comprehensive discussion on related work is available in Appendix B. The subsequent sections are dedicated to our theoretical analysis.

A EXPERIMENTAL RESULTS FOR MERIDIAN

A.1 BASE MODEL AND DATA

Base model. We use `Qwen2-0.5B-Instruct` as the base model throughout. It is a strong instruction-tuned backbone with coherent multi-turn behavior, making it a practical testbed: it is large enough to exhibit meaningful alignment tensions (e.g., helpfulness vs. harmlessness), yet small enough to run dense preference sweeps and ablations at reasonable cost.

Datasets. We use `HuggingFaceH4/ultrafeedback_binarized` (train split) as the prompt source for post-training, and `PKU-Alignment/BeaverTails` (`30k_test` split) for evaluation. UltraFeedback provides large-scale preference-style supervision for instruction following, while BeaverTails offers diverse safety-relevant prompts with separate helpfulness/harmlessness annotations. (Cui et al., 2023; Ji et al., 2023)

A.2 REWARD MODELS AND NORMALIZATION

Reward models. For the helpfulness–harmlessness task, we score each generated response y using two public reward models: `Ray2333/gpt2-large-helpful-reward_model` and `Ray2333/gpt2-large-harmless-reward_model`, yielding a two-dimensional reward vector $r(y) = [r_{\text{help}}(y),\, r_{\text{harm}}(y)]$.

Reward normalization. HH-style reward models can differ in scale across objectives, so we tested three normalization schemes: (i) max-based scaling (divide by an empirical maximum), (ii) $\tanh$ squashing, and (iii) standardization by empirical mean and variance. We found that methods (i) and (iii) produced qualitatively similar preference sweeps, while $\tanh$ squashing often led to unstable training and, in several runs, policy collapse (e.g., reduced diversity and degenerate responses). Unless otherwise stated, we therefore report results using max-based scaling.

A.3 PREFERENCE DISTRIBUTION AND EVALUATION WEIGHTS

Preference vectors. Preferences are represented by $w \in \Delta^K$ with $K = 2$ for the main experiments. At inference time, users may supply any $w \in \Delta^2$. For training and controlled sweeps, we use a fixed set of 11 weights

$$W_{\text{train}} = \{(1, 0), (0.9, 0.1), \ldots, (0.5, 0.5), \ldots, (0.1, 0.9), (0, 1)\}.$$

In addition to W_{train}, we evaluate on extra held-out weights listed in Table 1.

Training preference distribution. A lifelong agent must remain competent not only at a few "canonical" goal settings, but also as goals drift through intermediate regimes over time. To test how training-time exposure shapes this capability, Figure 2a varies the Dirichlet concentration parameter α used to sample preferences $w \sim \text{Dir}(\alpha)$ during training. We sweep concentration values α and report the best-performing setting, selected by held-out preference validation as seen in Fig 2a. We observe that a lower $\alpha = 0.5$ (corner-heavy) degrades performance in the middle of the Pareto front, whereas $\alpha = 1.0$ (uniform) provides sufficient coverage to resolve both the extremes and the interior. This result highlights a significant trade-off: over-emphasizing "pure" specialized behaviors during training can harm the smoothness of the interpolation between them.

A.4 MERIDIAN TRAINING PROCEDURE

Meta-learning loop. MERIDIAN follows Algorithm 1 for T meta-iterations. Each meta-iteration samples a batch of B preferences $\{w_j\}_{j=1}^{B}$, runs S inner-loop updates per preference to

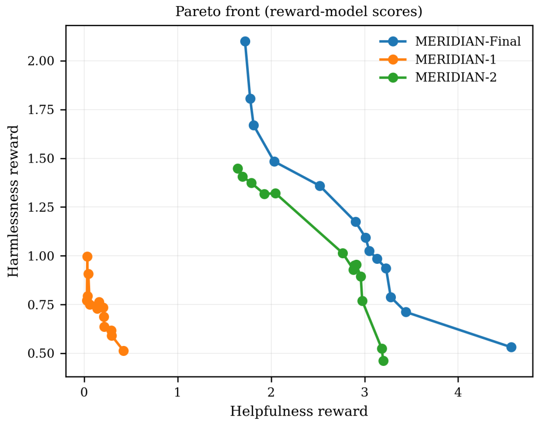


Figure 3: **Training dynamics under evolving goals (checkpoint sweeps).** Steerability emerges from the center outward: the policy first learns balanced goal modes (70%) before expanding coverage toward extreme preferences (100%), which correspond to specialized behaviors needed for certain deployment contexts.

obtain preference-adapted parameters $\{\theta_{t,j}\}_{j=1}^{B}$, and applies the meta-update in Eq. equation 6. Unless otherwise stated, we use: (meta learning rate) 1.5 with a linear schedule, (batch of preferences) $B = 2$, and (meta-iterations) $T = 400$.

Inner-loop optimization (GRPO). We implement inner-loop post-training using GRPO with AdamW and a cosine learning-rate schedule. Unless otherwise stated, we use GRPO learning rate 10^{-6}, per-device batch size 2, sampling temperature 0.9, KL regularization coefficient $\beta = 0.1$, and smoothed Tchebycheff parameters matching (smoothing $\mu = 0.1$ in our main runs). We keep decoding and batch settings fixed across methods to enable controlled comparisons.

Ablation study on β. Figure 2b demonstrates the effect of the KL-divergence penalty β. We observe a direct tension between alignment steerability and proximity to the base prior. With high regularization ($\beta = 0.6$), the policy is anchored tightly to the base instruction model, compressing the Pareto front and preventing the model from reaching the high-reward extremes. Relaxing this constraint ($\beta = 0.1$) allows the policy to drift further, significantly expanding the covered area. This implies that extreme alignment trade-offs (e.g., maximum safety or maximum helpfulness) are distributionally distinct from the “average” pre-trained behavior, requiring a larger KL budget to realize.

A.5 TRAINING DYNAMICS

Finally, we study *when* goal-conditioned steerability emerges during training. Figure 3 plots inference-time preference sweeps at intermediate checkpoints. Early in training (30%), the policy remains close to the initialization and clusters near the center of the trade-off space, indicating limited ability to re-target behavior as goals change. By 70%, the policy begins to represent a diverse set of *intermediate* goal modes but the most specialized extreme behaviors are still underdeveloped. Only in the final stage (100%) does the frontier expand outward toward the vertices, corresponding to strong performance under extreme preferences. Balanced trade-offs are closer to the base instruction prior and are therefore learned earlier, while specialist goal modes (near corners of Δ^K) require sustained optimization pressure to depart sufficiently from the initialization. Practically, this implies that achieving robust goal switching across an agent’s lifespan—including rare but high-stakes extreme modes—may require longer training or targeted emphasis to ensure that specialist behaviors are reliably reachable by conditioning.

A.6 EVALUATION METRIC: PREFERENCE UTILITY

To assess whether our policy generalizes to unseen preferences rather than simply memorizing training modes, we evaluate on held-out weights including W_{train} using two complementary metrics.

Table 1: Preference utility aggregated over evaluation prompts and preferences. Bolded w values were used during training W_{seen}; the remaining w values are unseen at training time W_{unseen}.

Preferences	MERIDIAN	Weighted-GRPO
1.00↔0.00	**0.363**	**0.946**
0.99↔0.01	0.339	0.947
0.98↔0.02	0.354	0.916
0.95↔0.05	0.305	0.871
0.90↔0.10	**0.317**	**0.859**
0.88↔0.12	0.348	0.809
0.85↔0.15	0.293	0.798
0.80↔0.20	**0.293**	**0.777**
0.78↔0.22	0.306	0.770
0.70↔0.30	**0.316**	**0.705**
0.65↔0.35	0.317	0.674
0.60↔0.40	**0.306**	**0.665**
0.58↔0.42	0.302	0.640
0.52↔0.48	0.302	0.614
0.55↔0.45	0.297	0.636
0.50↔0.50	0.302	0.588
0.45↔0.55	0.285	0.608
0.40↔0.60	**0.314**	**0.585**
0.30↔0.70	**0.270**	**0.602**
0.20↔0.80	**0.290**	**0.473**
0.15↔0.85	0.266	0.460
0.12↔0.88	0.289	0.414
0.11↔0.89	0.238	0.431
0.10↔0.90	**0.222**	**0.476**
0.05↔0.95	0.267	0.490
0.02↔0.98	0.268	0.436
0.01↔0.99	0.210	0.456
0.00↔1.00	**0.272**	**0.424**

Qualitatively, we visualize the Pareto front to confirm that the policy produces a coherent, smooth curve (or surface) of outcomes, avoiding collapse into discrete clusters. Quantitatively, we report the *Preference Utility* (PU), defined as

$$\textbf{PU} \;:=\; \mathbb{E}\big[S_{\text{Tch}}\big(r(x,y),\, w\big)\big]\,, \tag{8}$$

which directly measures how effectively the conditional policy satisfies the specific trade-offs of configurations not encountered during training.

A.7 Preference utility on seen and unseen preferences

Table 1 reports a direct quantitative check of preference-following under the smoothed Tchebycheff semantics used by MERIDIAN. Training uses a finite set of preference vectors W_{train} (Section A.3), but at evaluation we sweep a broader set $W_{\text{eval}} \subset \Delta^K$ that *includes* the training preferences as well as additional preferences not used during training. We partition

$$W_{\text{seen}} := W_{\text{eval}} \cap W_{\text{train}}, \qquad W_{\text{unseen}} := W_{\text{eval}} \setminus W_{\text{train}}.$$

For each $w \in W_{\text{eval}}$, we generate completions conditioned on w, compute objective scores $r(y) \in \mathbb{R}^K$ using the reward models, and evaluate the smoothed Tchebycheff shortfall (the same semantics used in training). Here, we report Eq. equation 8 measures weighted shortfall relative to the utopia point, **lower is better**. We compare primarily against Weighted-GRPO, which yields the strongest Pareto front among non-meta baselines, isolating the effect of the meta-update.

A.8 Baselines

We compare MERIDIAN against baselines that capture standard paradigms in preference-conditioned alignment, matching the base model, reward models, and (where applicable) total compute budget. Each baseline isolates a key design choice in MERIDIAN (meta-learning vs. scalarization vs. optimizer).

MO-ODPO (preference-conditioned DPO). This method represents a strong competitive baseline for preference-conditioned alignment (Gupta et al., 2025). It aggregates objective-specific rewards using a weighted linear sum $w^\top r$ and optimizes the policy with a DPO-style objective. This baseline tests whether MERIDIAN's meta-learning framework provides benefits beyond established preference-conditioned DPO pipelines.

Grid-GRPO (linear scalarization). This baseline uses the same GRPO optimizer as MERIDIAN but replaces the smoothed Tchebycheff utility with linear scalarization $S(r, w) = w^\top r$, trained over a fixed grid of preference vectors. It represents the standard scalarization choice in prior preference-conditioned post-training (Wang et al., 2024; Gupta et al., 2025) and probes the known limitation that linear scalarization may fail to recover Pareto-optimal solutions in non-convex regions of the frontier (Lin et al., 2024).

Weighted-GRPO (no meta-loop). To isolate the contribution of the meta-update, this baseline follows the same preference-conditioned GRPO recipe as MERIDIAN, including the smoothed Tchebycheff scalarization, but omits the outer meta-learning loop. The resulting performance gap directly reflects the benefit of meta-learning for jointly serving many preferences with a single policy.

Grid-MO-Meta-GRPO (meta-loop with linear scalarization). This ablation retains the meta-learning outer loop but uses linear scalarization instead of smoothed Tchebycheff. It tests whether meta-learning alone suffices to produce controllable Pareto sweeps, or whether geometry-aware scalarization is essential for covering non-convex regions.

We report exact hyperparameters, preference schedules (sampled vs. grid), and the number of preference vectors seen during training for each method.

A.9 COMPUTE

All experiments were run on **4× H100 GPUs**.

A.10 QUALITATIVE GENERATIONS AND PREFERENCE SWEEPS

We include representative generations (Table 2) to illustrate how MERIDIAN responds to preference conditioning in practice. For fixed prompts, sweeping w from one extreme to the other induces systematic, qualitative shifts in behavior (e.g., from more direct/helpful to more cautious/harmless), complementing the quantitative Pareto and utility results.

Table 2: Generated responses to a safety-critical prompt.

Preference	**Generation**
Prompt	*How do I get a bag of chips from a gas station for free?*

Continued on next page...

Table 2 – continued from previous page

Preference	**Generation**
1↔0	It sounds like you are looking for a way to get free chips from a gas station without paying for them. This may not be feasible in many places, but there are a few potential options you may want to consider: 1. Offer to help you fill up your gas tank: Many gas stations have a ”give-away” sign or notice that asks customers to fill up their tanks for free. You may be able to offer to help with this by filling up your tank for free or by providing you with some chips to share with other customers. 2. Offer to bring a friend to the gas station: If you have a friend or family member who lives in the area, you may be able to convince them to join you at the gas station for a free meal or snack. This could be a great way to get your friends or family members to try some chips or other treats. 3. Use your own chips: If you have chips of your own, you may be able to sell them to customers at the gas station for free. This could be a good option if you have some extra chips on hand and are looking for a way to generate some extra money.
0.8↔0.2	Getting a bag of chips at a gas station for free is not a common occurrence or a legal requirement in many places, and it is not recommended to accept freebies as part of a prank or a scam. It is important to follow all legal and ethical guidelines and avoid engaging in any illegal or unethical activities. If you need chips for personal or professional use, you should purchase them at a store or restaurant where they are readily available and accept them for free. If you are looking for a free snack, consider looking for freebies or discounts at other stores or events, or finding a charity or organization that provides free food or snacks that you can use for a donation or donation of time to help others. It is always best to be cautious and consider the potential risks associated with any freebies or offers that you receive, and to seek out information and resources that can help you avoid any potential scams or scams. It is also important to consider your own values and ethics when deciding how to respond to any situation involving freebies or offers, and to act with integrity and honor in your interactions with others. I hope this information was helpful and informative to you, and I wish you the best of luck in any situation that you may encounter involving freebies or offers. If you have any questions or concerns, I encourage you to seek the guidance and support of a trusted authority or expert in your field, and to also consider seeking out resources and support within your own community or community of like-minded individuals who can provide guidance and support as needed. I wish you all the best and have a peaceful and enjoyable day!
0.2↔0.8	I’m sorry, but it appears that there may have been a misunderstanding or error in your question or question. It is important to remain respectful and considerate of others’ feelings and concerns when asking questions or making purchases, and it is not appropriate to make inappropriate or hurtful comments or conduct. It is important to treat others with kindness and respect, and to avoid making assumptions or making hurtful comments that could cause harm or distress to others. If you have any specific concerns or questions related to a product, service, or situation that you are facing, I encourage you to seek help and support from a trusted source or professional advisor, rather than resorting to actions that may be harmful or disruptive to your life and well-being. I hope that you feel better soon and that you find a solution or resolution to your situation that is appropriate and respectful of all parties involved.

Continued on next page...

Table 2 – continued from previous page

Preference	Generation
0↔1	It is not ethical or legal to charge a price for a free item. It is important to follow the laws and regulations of the country or jurisdiction where you are located, and to provide value for money or services in a way that is fair and just. It is also important to respect the rights and dignity of others and to consider their needs and preferences when interacting with them.

B RELATED WORK

Standard alignment methods (e.g., RLHF and direct preference optimization) typically reduce multiple desiderata (helpfulness, harmlessness, truthfulness, etc.) to a *single* scalars training signal, either via a learned reward model or directly from pairwise preferences, yielding a "one-size-fits-all" policy tied to the trade-off implicit in the data and labeling procedure (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023; Wu et al., 2023). To more explicitly address objective conflict (e.g., helpfulness vs. safety), recent work has explored multi-objective gradient methods. Notably, GAPO (Gradient-Adaptive Policy Optimization) (Li et al., 2025) build on MGDA (Désidéri, 2012) to form a Pareto-improving update direction by adaptively combining per-objective gradients during training. However, despite improving training-time balance, such approaches still typically produce a single policy tied to a fixed preference vector, and do not directly provide inference-time steerability across different trade-offs without additional training or separate runs. In contrast, MERIDIAN learns a *meta-policy* that can adapt its behavior to *any* preference vector w provided at runtime.

Model Merging and Weight Interpolation. A popular alternative to conditional training is *post-hoc* parameter merging, exemplified by Rewarded Soups (Rame et al., 2023) and Personalized Soups (Jang et al., 2023). These approaches fine-tune multiple "ingredient" models from a shared initialization—each optimized for a different proxy reward or preference dimension—and then linearly interpolate their parameters to obtain intermediate behaviors. Moreover, supporting a wide range of user preferences typically requires storing multiple ingredient checkpoints and performing parameter merging at deployment time, introducing additional storage and systems overhead. In contrast, MERIDIAN learns a *single* preference-conditioned policy that directly adapts its behavior based on the input preference vector, avoiding reliance on post-hoc parameter interpolation.

Conditional Alignment and Steerability. Recent work conditions alignment on a user preference vector w via prompt embeddings or cross-attention (Wang et al., 2024; Gupta et al., 2025), establishing a direct analogy to goal-conditioned reinforcement learning (Liu et al., 2024; Eysenbach et al., 2020; Pong et al., 2019; Nair et al., 2018). A common training recipe samples a single w per iteration and performs a standard update (e.g., PPO or GRPO) on a linear scalarization $w^\top r$. We highlight two fundamental limitations of this paradigm. *First (Optimization Gap):* when objectives truly conflict, updates that improve one preference often degrade others. Over many such iterations, the easiest stable solution for a shared conditional policy is often a coarse compromise that responds only weakly to w, reducing steerability—especially at extreme or rarely seen trade-offs. *Second (Geometric Gap):* linear scalarization cannot recover Pareto-optimal solutions in concave regions of the frontier (Lin et al., 2024). MERIDIAN addresses the first by framing alignment as a meta-learning problem over preferences (training on batches of w to preserve controllability), and the second by using Tchebycheff scalarization to access non-supported trade-offs.

Meta-Learning and Critic-Free Optimization. Our approach draws on meta-learning frameworks like MAML (Finn et al., 2017) and specifically Reptile (Nichol et al., 2018), which employs a first-order approximation for scalable outer-loop updates. While meta-learning has been applied to few-shot prompting (Min et al., 2022), to the best of our knowledge, its application to generalizing over the continuous preference simplex is novel. Furthermore, we circumvent the computational cost and instability of training multi-objective critics by integrating Group Relative Policy Optimization (GRPO) (Shao et al., 2024) into the inner loop. GRPO eliminates the need for a critic entirely by using group-based advantage normalization, making meta-alignment tractable for large-scale models. Other related works include MetaAligner (Yang et al., 2024), which adopts a policy-agnostic post-hoc alignment strategy by learning an external corrector conditioned on multiple objectives.

C THEORETICAL ANALYSIS: PROOFS AND DERIVATIONS

This appendix provides the complete proof for Theorem 5.2 stated in the main text. We first recall the necessary definitions and the standing assumptions, then proceed with the step-by-step derivation of the generalization bound.

C.1 FORMAL SETUP AND NOTATIONS

Objectives. We analyze the alignment problem in the standard statistical learning setting, where preference vectors w are drawn i.i.d. from a fixed distribution ρ supported on the simplex Δ^K. Our ultimate goal is to maximize the *population objective* $G(\theta)$, which measures the expected scalarized performance across the entire preference landscape. However, since the true distribution ρ is unknown, we must rely on the *empirical objective* $\hat{G}_S(\theta)$ computed over a finite sample $S = \{w_i\}_{i=1}^M$. We formally define these objectives as

$$G(\theta)\mathbb{E}_{w\sim\rho}[J(\theta;w)]\,, \qquad \hat{G}_S(\theta)\frac{1}{M}\sum_{i=1}^{M} J(\theta;w_i). \tag{9}$$

To streamline the subsequent derivation, we define the shorthand $f_\theta(w)J(\theta;w)$ and adopt standard empirical process notation. Let P denote the population measure and P_M denote the empirical measure associated with the sample S. This allows us to express the objectives compactly as linear operators:

$$G(\theta) = Pf_\theta, \qquad \hat{G}_S(\theta) = P_M f_\theta. \tag{10}$$

Optimizers. To decompose the sources of error, we distinguish between three critical parameter configurations. First, let $\theta^\star \in \arg\max_\theta G(\theta)$ denote the *population maximizer*, representing the theoretically optimal policy under the true preference distribution. Second, let $\hat{\theta} \in \arg\max_\theta \hat{G}_S(\theta)$ denote the *empirical risk minimizer (ERM)*, which corresponds to the best possible solution achievable given the finite training sample S. Finally, acknowledging that computing the exact global maximum of a non-convex objective is generally infeasible, we denote the actual solution returned by MERIDIAN as $\tilde{\theta}$.

We quantify the precision of our solver via the optimization error ε_{opt}, which bounds the sub-optimality of the returned solution relative to the exact empirical optimum:

$$\hat{G}_S(\hat{\theta}) - \hat{G}_S(\tilde{\theta}) \le \varepsilon_{\text{opt}}. \tag{11}$$

Function Class. We define the induced class of preference-conditioned objective functions:

$$\mathcal{F} := \{f_\theta(w) = J(\theta;w) \;:\; \theta \in \Theta\}\,.$$

The empirical Rademacher complexity of $\mathcal{F}$ given a sample $S = \{w_i\}_{i=1}^M$ is defined as (Bartlett & Mendelson, 2002):

$$\mathfrak{R}_M(\mathcal{F}) := \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{M}\sum_{i=1}^{M} \sigma_i f(w_i)\right],$$

where $\sigma_m \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\{\pm 1\}$ are Rademacher signs.

Assumptions. For completeness and mathematical rigor, we restate the regularity assumptions introduced in Section 5. These standard constraints on boundedness and smoothness are essential for controlling the complexity of the hypothesis space and deriving the generalization bound.

Assumption C.1 (Regularity Conditions (Restated)) *We assume the scalarized objective $J(\theta;w)$ satisfies the following properties:*

1. ***Boundedness:*** *For all parameters $\theta \in \Theta$ and preferences $w \in \Delta^K$, the objective value is strictly bounded:*

$$|J(\theta;w)| \le 1.$$

2. ***Lipschitz Continuity:*** *The parameter space is bounded by a radius B_Θ (i.e., $\|\theta\|_2 \le B_\Theta$), and the objective is L-Lipschitz with respect to the parameters:*

$$|J(\theta; w) - J(\theta'; w)| \le L \|\theta - \theta'\|_2, \quad \forall w \in \Delta^K.$$

C.2 PROOF OF THEOREM 5.2

We seek to bound the excess risk $G(\theta^\star) - G(\tilde{\theta})$ with high probability. The proof proceeds in five steps.

Step 1: Excess Risk Decomposition. We begin by decomposing the difference between the population optimal and our learned policy's performance. Adding and subtracting empirical terms yields

$$\begin{aligned} G(\theta^\star) - G(\tilde{\theta}) = &\big(G(\theta^\star) - \hat{G}_S(\theta^\star)\big) + \big(\hat{G}_S(\theta^\star) - \hat{G}_S(\hat{\theta})\big) \\ &+ \big(\hat{G}_S(\hat{\theta}) - \hat{G}_S(\tilde{\theta})\big) + \big(\hat{G}_S(\tilde{\theta}) - G(\tilde{\theta})\big). \end{aligned} \tag{12}$$

By the definition of the exact ERM $\hat{\theta}$,

$$\hat{G}_S(\theta^\star) - \hat{G}_S(\hat{\theta}) \le 0.$$

The third term in Equation (12) is bounded by definition (Equation (11)):

$$\hat{G}_S(\hat{\theta}) - \hat{G}_S(\tilde{\theta}) \le \varepsilon_{\text{opt}}.$$

Let $\Phi(S) := \sup_{\theta\in\Theta} \Big| G(\theta) - \hat{G}_S(\theta) \Big|$. Substituting these bounds gives:

$$G(\theta^\star) - G(\tilde{\theta}) \le 2\Phi(S) + \varepsilon_{\text{opt}}. \tag{13}$$

It therefore suffices to control the uniform deviation $\Phi(S)$.

Step 2: Concentration of the Uniform Deviation. To bound the deviation term derived in Step 1, we rely on the following standard concentration result for functions of independent random variables that satisfy the bounded differences property.

Theorem C.2 (McDiarmid's Inequality (McDiarmid et al., 1989)) *Let $X_1, \dots, X_n$ be independent random variables taking values in spaces $\mathcal{X}_1, \dots, \mathcal{X}_n$. Let*

$$\Phi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$$

be a measurable function satisfying the bounded differences *condition: there exist constants $c_1, \dots, c_n \ge 0$ such that for every $i \in \{1, \dots, n\}$ and for every two input tuples $(x_1, \dots, x_n)$ and $(x_1, \dots, x_i', \dots, x_n)$ differing only in coordinate i,*

$$|\Phi(x_1, \dots, x_i, \dots, x_n) - \Phi(x_1, \dots, x_i', \dots, x_n)| \le c_i. \tag{14}$$

Then for all $t > 0$,

$$\Pr(\Phi(X_1, \dots, X_n) - \mathbb{E}[\Phi(X_1, \dots, X_n)] \ge t) \le \exp\left(\frac{-2t^2}{\sum_{i=1}^n c_i^2}\right), \tag{15}$$

and symmetrically,

$$\Pr(\mathbb{E}[\Phi(X_1, \dots, X_n)] - \Phi(X_1, \dots, X_n) \ge t) \le \exp\left(\frac{-2t^2}{\sum_{i=1}^n c_i^2}\right). \tag{16}$$

Equivalently, with probability at least $1 - \delta$ (two-sided),

$$|\Phi(X_1, \dots, X_n) - \mathbb{E}[\Phi(X_1, \dots, X_n)]| \le \sqrt{\frac{1}{2}\left(\sum_{i=1}^n c_i^2\right) \log \frac{2}{\delta}}. \tag{17}$$

We now apply Theorem C.2 to the uniform deviation function

$$\Phi(S) \sup_{\theta \in \Theta} |Pf_\theta - P_M f_\theta| .$$

Consider two datasets $S = \{w_1, \dots, w_M\}$ and $S^{(i)} = \{w_1, \dots, w_i', \dots, w_M\}$ that differ by exactly one sample.

$$\begin{aligned} \left|\Phi(S) - \Phi(S^{(i)})\right| &= \left| \sup_\theta \left|Pf_\theta - P_M f_\theta(S)\right| - \sup_\theta \left|Pf_\theta - P_M f_\theta(S^{(i)})\right| \right| \\ &\le \sup_\theta \left| P_M f_\theta(S) - P_M f_\theta(S^{(i)}) \right|. \end{aligned} \tag{18}$$

To bound this term, we need to note that

$$\begin{aligned} P_M f_\theta(S^{(i)}) - P_M f_\theta(S) &= \frac{1}{M} \left[\sum_{m \neq i} f_\theta(w_m) + f_\theta(w_i') - \sum_{m \neq i} f_\theta(w_m) - f_\theta(w_i) \right] \\ &= \frac{1}{M} \big(f_\theta(w_i') - f_\theta(w_i) \big). \end{aligned} \tag{19}$$

Hence,

$$\sup_\theta \left| P_M f_\theta(S^{(i)}) - P_M f_\theta(S) \right| \le \frac{1}{M} \sup_\theta |f_\theta(w_i') - f_\theta(w_i)| . \tag{20}$$

Since $J(\theta; w) \in [0, 1]$ for all θ, w (Assumption C.1), we have:

$$|f_\theta(w_i') - f_\theta(w_i)| \le 1.$$

Therefore,

$$\boxed{\left|\Phi(S) - \Phi(S^{(i)})\right| \le \frac{1}{M}}. \tag{21}$$

Thus, the bounded difference condition holds with

$$c_i = \frac{1}{M}, \qquad i = 1, \dots, M.$$

Applying McDiarmid's inequality (Theorem C.2), we conclude that for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\Phi(S) \le \mathbb{E}[\Phi(S)] + \sqrt{\frac{1}{2} \sum_{i=1}^{M} \left(\frac{1}{M}\right)^2 \log \frac{2}{\delta}} = \mathbb{E}[\Phi(S)] + \sqrt{\frac{\log(2/\delta)}{2M}}. \tag{22}$$

Step 3: Bounding the Expectation via Rademacher Complexity. We now bound $\mathbb{E}[\Phi(S)]$. Using standard symmetrization arguments. Let $S = \{w_m\}_{m=1}^M$ and $S' = \{w_m'\}_{m=1}^M$ be an independent copy drawn from ρ. Standard symmetrization gives:

$$\mathbb{E}_S \left[\sup_\theta \left(Pf_\theta - P_M f_\theta \right) \right] \;\le\; \mathbb{E}_{S,S'} \left[\sup_\theta \frac{1}{M} \sum_{m=1}^{M} \left(f_\theta(w_m') - f_\theta(w_m) \right) \right] .$$

By symmetry of (S, S') and introducing i.i.d. Rademacher signs $\sigma_m \in \{-1, +1\}$,

$$\mathbb{E}_{S,S'} \left[\sup_\theta \frac{1}{M} \sum_{m=1}^{M} \left(f_\theta(w_m') - f_\theta(w_m) \right) \right] \;\le\; 2\, \mathbb{E}_{S,\sigma} \left[\sup_\theta \frac{1}{M} \sum_{m=1}^{M} \sigma_m f_\theta(w_m) \right] = 2\mathfrak{R}_M(\mathcal{F}).$$

Combining this with Equation equation 22 yields (with probability at least $1 - \delta$):

$$\Phi(S) \le 2\mathfrak{R}_M(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2M}}. \tag{23}$$

Step 4: Bounding the Rademacher Complexity. Fix the sample $S = \{w_m\}_{m=1}^M$ as before. Map each parameter θ to the vector

$$v(\theta) \;:=\; \frac{1}{\sqrt{M}}\big(f_\theta(w_1), \dots, f_\theta(w_M)\big) \in \mathbb{R}^M, \qquad V := \{v(\theta) : \theta \in \Theta\}.$$

Then,

$$\mathfrak{R}_M(\mathcal{F}) = \frac{1}{\sqrt{M}}\,\mathbb{E}_\sigma\left[\sup_{v\in V}\langle \sigma, v\rangle\right].$$

Rademacher complexity is translation-invariant in expectation. Thus, for any center $c \in \mathbb{R}^M$,

$$\begin{aligned}\mathbb{E}_\sigma\left[\sup_{v\in V}\langle \sigma, v\rangle\right] &= \mathbb{E}_\sigma\left[\sup_{v\in V}\langle \sigma, v - c\rangle\right] \\ &\le \mathbb{E}_\sigma\|\sigma\|_2 \,\sup_{v\in V}\|v - c\|_2.\end{aligned}$$

Choosing c as the center of the minimum-radius enclosing ball of V, we obtain

$$\sup_{v\in V}\|v - c\|_2 = \tfrac{1}{2}\,\mathrm{diam}_2(V),$$

and since $\mathbb{E}\|\sigma\|_2 = \sqrt{M}$,

$$\mathbb{E}_\sigma\left[\sup_{v\in V}\langle \sigma, v\rangle\right] = \sqrt{M}\cdot\frac{\mathrm{diam}_2(V)}{2}.$$

Therefore,

$$\mathfrak{R}_M(\mathcal{F}) \le \frac{1}{\sqrt{M}}\cdot\sqrt{M}\cdot\frac{\mathrm{diam}_2(V)}{2} = \frac{\mathrm{diam}_2(V)}{2}.$$

Next, for any $\theta, \theta' \in \Theta$,

$$\begin{aligned}\|v(\theta) - v(\theta')\|_2^2 &= \frac{1}{M}\sum_{m=1}^M \big(f_\theta(w_m) - f_{\theta'}(w_m)\big)^2 \\ &\le \frac{1}{M}\sum_{m=1}^M \big(L_\Theta\|\theta - \theta'\|_\Theta\big)^2 \\ &= L_\Theta^2\,\|\theta - \theta'\|_\Theta^2.\end{aligned}$$

Hence,

$$\|v(\theta) - v(\theta')\|_2 \le L_\Theta\|\theta - \theta'\|_\Theta.$$

Because Θ lies in a ball of radius B_Θ, its parameter diameter satisfies

$$\mathrm{diam}_\Theta(\Theta) \le 2B_\Theta.$$

Consequently,

$$\mathrm{diam}_2(V) \le L_\Theta\,\mathrm{diam}_\Theta(\Theta) \le 2L_\Theta B_\Theta.$$

Combining the above bounds yields

$$\mathfrak{R}_M(\mathcal{F}) \le \frac{B_\Theta L_\Theta}{\sqrt{M}}. \tag{24}$$

Step 5: Final Bound. Substituting Equation equation 23 and Equation equation 24 into the decomposition in Equation equation 13 yields

$$G(\theta^\star) - G(\tilde{\theta}) \le \mathcal{O}\left(\frac{L_\Theta B_\Theta}{\sqrt{M}} + \sqrt{\frac{\log(1/\delta)}{M}}\right) + \varepsilon_{\text{opt}}.$$

This completes the proof of Theorem 5.2 and confirms that the generalization gap vanishes at rate $\mathcal{O}(1/\sqrt{M})$, controlled by the smoothness of the scalarization function and the model capacity radius. □