ROTATION-BASED GRADIENT BALANCING: RECONCILING ALIGNMENT AND FIDELITY IN MULTI-TASK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task learning (MTL) enables knowledge sharing across tasks but often suffers from gradient conflicts. Gradient manipulation methods reduce these conflicts by directly altering task gradient angles to eliminate opposing directions. However, over-prioritizing geometric alignment may introduce substantial deviations from the original gradient directions, diluting valuable task-specific information and degrading performance. To overcome these limitations, we propose the Rotation-Based Gradient Balancing (RGB) method. RGB enables task gradients to be finely adjusted toward a consensus direction, holistically reducing global conflict while penalizing excessive deviation from the original task-gradient directions. Our method consistently achieves competitive results across various MTL benchmarks ranging from 3 to 40 tasks. Empirically, ablation studies show that simply increasing the degree of rotation does not consistently improve task performance, reinforcing the importance of controlling gradient modification when balancing alignment and fidelity.

1 INTRODUCTION

Multi-task learning (MTL) is a paradigm that enables a single model to simultaneously optimize multiple tasks given a set of shared parameters. It promotes knowledge transfer between tasks to improve the model's generalization and reduce computational overhead without the need to train separate redundant models for practical applications. However, MTL often suffers from the gradient conflict issue, where the task-specific gradients are in opposing directions. Improper aggregation of these gradients can lead to suboptimal updates, causing performance imbalances where dominant tasks suppress the learning of others.

Existing MTL approaches can be broadly categorized into weighting-indirect balancing and gradient-direct balancing methods to resolve the corresponding issues. The former relies on loss or gradient information to adjust each task's contribution to the shared update. Representative methods include MGDA-UB Sener & Koltun (2018), CAGrad Liu et al. (2021a), Nash-MTL Navon et al. (2022), FAMO Liu et al. (2023), and FairGrad Ban & Ji (2024). For example, FairGrad formulates task balancing through an α-fair utility of first-order task progress, constructing a shared update that reflects different preferences for fairness across tasks. Although these methods can mitigate task interference by changing the aggregate update direction, positive scalar reweighting preserves the pairwise cosine similarities between task gradients and therefore does not directly modify their pairwise conflict geometry.

Alternatively, gradient manipulation methods provide a stronger remedy for gradient conflict by directly altering the task's gradient Zhang et al. (2024). For instance, PCGrad Yu et al. (2020) relies on orthogonal projections to subtract the conflicting component of one task's gradient from another in a pairwise manner before unifying the update direction. GradDrop Chen et al. (2020) probabilistically drops the gradient signs based on their alignment across tasks to reduce their interference during backpropagation. GradVac Wang et al. (2021) adjusts conflicting gradients through pairwise projections based on cosine similarities relative to exponentially moving averaged historical correlations. RotoGrad Javaloy & Valera (2022) reduces the gradient conflict by introducing a learning-based rotation parameter to transform task gradients for alignment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing gradient-direct balancing methods face some major challenges that limit their performance. We summarize these limitations as follows: 1) Rigorous projection mechanisms may over-react in scenarios of marginal conflict, over-penalizing the task gradient; 2) Iterative manipulation of gradient pairs in locally pairwise manner fails to capture the global task's conflict status; and 3) Over-prioritize geometric alignment without considering the cost of erasing valuable task-specific information.

First, gradient-direct balancing methods often suffer from performance inconsistency due to their rigorous correction. For example, in PCGrad, for gradients g_i and g_j with negative cosine similarity (angles exceeding 90°), g_i is updated as $g_i \leftarrow g_i - \frac{g_i^\top g_j}{\|g_j\|^2} g_j$, fully removing the conflicting component to ensure non-negative alignment. GradDrop might potentially drop important gradient information. In the case of minor conflicts, where cosine similarity is only marginally negative. The complete elimination of the opposing component may over-penalize gradients, potentially leading to suboptimal update directions and degrading the task performance.

Secondly, current gradient-direct balancing methods are mainly conducted in a localized, pairwise manner with random order before aggregation. For T tasks, this requires processing up to $\binom{T}{2}$ pairs, imposing quadratic computational scaling (e.g., 3 tasks involve 3 pairs; 4 tasks involve 6 pairs), that becomes a bottleneck in high-task regimes. Pairwise manipulation resolves local conflicts but lacks a holistic view of the gradient system, potentially disrupt the overall task-interrelation. For example, the projections are executed sequentially in a shuffled order, introducing order-dependence issue: earlier adjustments may be disrupted by subsequent modifications and causing bias in final task's gradients.

Thirdly, existing gradient manipulation methods, whether based on projection or rotation, primarily focus on geometric conflict reduction without concerning the cost of erasing valuable task-specific information. For example, RotoGrad optimizes its rotation parameters solely to minimize gradient conflict, which can lead to excessive alignment that overly constrains the shared representation and limits positive transfer.

Existing gradient manipulation methods often yield suboptimal performance due to a lack of theoretical grounding regarding the ideal adjustment magnitude. This raises a fundamental question: "How should gradient modification balance inter-task alignment with fidelity to the original task-gradient directions?". To address this, we propose a novel gradient manipulation algorithm, namely Rotation-Based Gradient Balancing (RGB), that jointly optimizes rotation angles vector α to finely adjust the task-gradient towards a consensus direction, explicitly balancing maximal conflict reduction with minimal distortion of task-specific information. Whereas FairGrad emphasizes fairness in task progress, RGB explicitly controls the geometric trade-off between alignment and fidelity.

Unlike RotoGrad, which exhibits architectural dependency–being designed specifically for hard-parameter-sharing setups and thus inapplicable to other MTL paradigms (e.g., soft sharing, cross-stitch networks, or fully separate modules) without modification. RGB operates purely at the optimization level and requires no additional rotation layers in the network, distinguishing its implementation from RotoGrad's learned rotations of shared representations.

Our contributions are threefold:

- We introduce RGB, an optimization-level gradient manipulation method for MTL that jointly optimizes task-specific rotation angles to balance global gradient alignment with fidelity to the original task directions.
- We formulate the Multi-Task Equilibrium Relationship (MER; Eq. 4) that is supported by our empirical analysis. Under Lipschitz smoothness, lower-bounded task losses, Robbins–Monro stepsizes, and a quantitative common descent condition, we prove that the Pareto-stationarity residual of the deterministic RGB update converges to zero and that every accumulation point is Pareto stationary (Appendix E).
- We demonstrate the effectiveness and consistency of our method, achieving competitive performance across various datasets ranging from 3-40 tasks, as detailed in Section 4.

2 PRELIMINARIES

2.1 PARETO OPTIMALITY, COMMON DESCENT DIRECTION, AND PARETO STATIONARITY

This section introduces the MTL fundamentals, where Pareto optimality refers to a point at which neither task can be improved without worsening another, and common descent direction refers to a direction in the unified parameter space that simultaneously reduces all task losses. Pareto stationarity refers to a point where no such common descent direction exists to improve all objectives at once. The MTL objective is to optimize a vector of the objective functions. Let $\theta \in \Theta = \mathbb{R}^D$ denote the shared model parameters, and let $F(\theta) = (f_1(\theta), \ldots, f_T(\theta))$ denote the vector of task-specific losses.

Definition 2.1 (Pareto Optimality)**.** A point $\theta^* \in \Theta$ is called *Pareto optimal* if there does not exist another feasible point $\theta \in \Theta$ such that $f_i(\theta) \leq f_i(\theta^*)$ for all $i \in \{1, \ldots, T\}$ and $F(\theta) \neq F(\theta^*)$. The set of all Pareto optimal solutions is called the *Pareto set*, and the corresponding objective vectors $\{F(\theta^*)\}$ are regarded as the *Pareto front*.

Definition 2.2 (Common Descent Direction)**.** A vector $v \in \mathbb{R}^D$ is called a *common descent direction* at a point θ if it simultaneously decreases all task losses at θ, i.e.,

$$\langle \nabla f_i(\theta), v \rangle < 0 \quad \forall i \in \{1, \ldots, T\}.$$

Definition 2.3 (Pareto Stationarity)**.** A point $\theta \in \Theta$ is called *Pareto stationary* if no common descent direction exists for all objectives at θ. Formally, this holds when

$$\operatorname{range}(\nabla F(\theta)^\top) \cap \left(-\mathbb{R}^T_{++}\right) = \emptyset,$$

where $\nabla F(\theta) = (\nabla f_1(\theta), \nabla f_2(\theta), \ldots, \nabla f_T(\theta)) \in \mathbb{R}^{D \times T}$ is the Jacobian of $F(\theta)$, and $\mathbb{R}^T_{++}$ denotes the positive orthant cone. When all task losses are convex, every Pareto stationary point is weakly Pareto optimal: no feasible point strictly improves all task losses simultaneously.

2.2 MULTI-TASK COST-BENEFIT MODEL

We write the shared update as $\theta - \eta v$, where v is the aggregated gradient vector. Accordingly, $-v$ is a common descent direction when $\langle g_i, v \rangle > 0$ for every task. We examine how gradient modification affects task progress through a first-order analysis of the shared update. Let $g_i \triangleq \nabla f_i(\theta)$ denote the gradient of task i. For task losses with Lipschitz-continuous gradients,

$$f_i(\theta - \eta v) = f_i(\theta) - \eta \langle g_i, v \rangle + R_i, \qquad |R_i| \leq \frac{\beta}{2} \eta^2 \|v\|^2.$$

for a fixed, bounded update vector v, with the remainder bound following from an explicit integral form (Appendix F.1). For the scalar analysis, let $\alpha \in [0, \pi/2]$ denote a common rotation amount, measured in radians, and let $r_i(\alpha)$ denote the transformed direction of task i. The corresponding aggregated vector is $v = \frac{1}{T} \sum_{i=1}^{T} r_i(\alpha)$. Let $\bar{f}(\alpha)$ denote the first-order approximation to the average loss after the shared update at a fixed θ and stepsize η. Separating the task's own contribution from the cross-task contributions gives the decomposition derived in Appendix G.1:

$$\bar{f}(\alpha) = \bar{f}(0) - \mathrm{CE}(\alpha) + \mathrm{FE}(\alpha). \tag{1}$$

Here, $\bar{f}(0)$ is the baseline first-order mean loss without rotation. The conflict effect $\mathrm{CE}(\alpha)$ measures the signed gain in cross-task first-order contributions, while the fidelity effect $\mathrm{FE}(\alpha)$ measures the signed loss in each task's own contribution. This formulation explicitly illustrates the optimization trade-off governed by the rotation angle α. When $r_i(\alpha)$ is performed with α rotation transformation toward consensus direction, the conflicting gradient issue is potentially mitigated with a better task alignment, meanwhile α deviation cost incurred when $r_i(\alpha)$ are deviating from their task-specific descent directions g_i.

Our angular sweeps experiment on Cityscapes, NYUv2, and QM9 reveal dataset-dependent preferences for gradient rotation (Appendix I). At later training checkpoints, Cityscapes exhibits a broad U-shaped relationship between rotation magnitude and five-step mean loss, whereas NYUv2 and QM9 favor small or near-zero rotations. Increasing rotation beyond these preferred ranges generally raises the loss. These observations motivate a scalar cost–benefit model of the competing effects of alignment and deviation.

Conflict Effect (CE). Negative inner products between task gradients indicate conflicting directions Yu et al. (2020). Adjusting these directions toward agreement can reduce cancellation in the shared update (Appendix B). To capture the diminishing gains from additional alignment, we model the benefit as

$$B(\alpha) = b\log(1+\alpha), \qquad b > 0, \tag{2}$$

where α is the rotation amount in radians and b controls the benefit scale. Its marginal benefit $B'(\alpha) = b/(1+\alpha)$ decreases with the rotation amount. This form captures the intuition that early alignment yields substantial gains, while additional alignment provides progressively smaller improvements.

Fidelity Effect (FE). Gradient manipulation—whether via projection or rotation inevitably deviates the updated direction from the original task-specific descent direction. Small deviations are often tolerable, acting as a form of regularization that mitigates noise or minor conflicts. However, as the degree of manipulation increases, the distortion grows, progressively erasing valuable task-specific information and impairing individual task optimization. In extreme cases, excessive modification can produce a shared update that is adversely aligned with an original task gradient, potentially increasing that task's loss. We represent an increasing marginal penalty for deviation through the cost model

$$C(\alpha) = c\big(\exp(k\alpha) - 1\big), \qquad c, k > 0, \tag{3}$$

where c controls the cost scale and k controls its growth rate. This formulation reflects the accelerating degradation caused by large deviations from task-specific gradients. Together, B and C express the benefit–cost trade-off through the net gain $B(\alpha) - C(\alpha)$.

2.3 Multi-Task Equilibrium Relationship (MER) and Geometrical Regimes

We characterize the optimal balance in the cost–benefit model by

$$\alpha^\star \in \arg\min_{\alpha\in[0,\pi/2]} \big\{C(\alpha) - B(\alpha)\big\}.$$

For Eqs. 2 and 3,

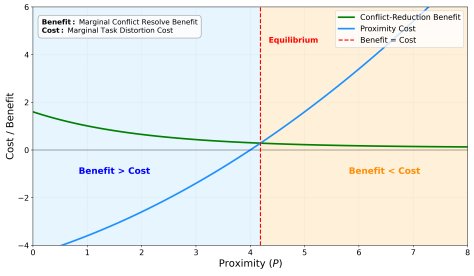
$$\frac{d^2}{d\alpha^2}\big(C(\alpha) - B(\alpha)\big) = ck^2\exp(k\alpha) + \frac{b}{(1+\alpha)^2} > 0,$$

so the minimizer is unique. When $b > ck$ and $b/(1+\pi/2) < ck\exp(k\pi/2)$, it lies in the interior of the interval and satisfies

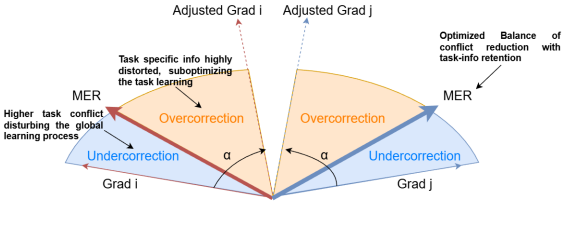
$$B'(\alpha^\star) = C'(\alpha^\star), \qquad \frac{b}{1+\alpha^\star} = ck\exp(k\alpha^\star). \tag{4}$$

Otherwise, the minimum occurs at a boundary (Theorem G.3).

We refer to the marginal balance in Eq. 4 as the Multi-Task Equilibrium Relationship (MER): at an interior optimum, the marginal benefit of alignment equals the marginal cost of fidelity deviation.



(a) Cost–benefit model. Gradient alignment modeled as a trade-off between benefit (green) and cost (blue).



(b) Geometric interpretation. The optimal gradient rotation delimiting the under-correction and over-correction regimes.

Figure 1: Multi-Task Equilibrium Relationship (MER). The scalar cost–benefit model characterizes the balance between gradient alignment and deviation from the original task directions.

As illustrated in Figure 1a, MER identifies the optimal rotation amount within the scalar cost–benefit model, where the marginal alignment benefit equals the marginal deviation cost. Figure 1b illustrates the two geometric regimes around an interior optimum:

Under-Correction Regime (Blue Region). For $\alpha < \alpha^\star$, the marginal benefit exceeds the marginal cost, $B'(\alpha) > C'(\alpha)$. In this regime, the rotation is conservative. Increasing rotation therefore improves the modeled net gain, indicating scope for further alignment.

Over-Correction Regime (Orange Region). For $\alpha > \alpha^\star$, the marginal cost exceeds the marginal benefit, $C'(\alpha) > B'(\alpha)$. In this regime, the modeled deviation cost grows faster than the alignment benefit. Further rotation therefore reduces the modeled net gain.

MER motivates controlling the amount of gradient modification. We implement this principle in RGB by jointly optimizing task-specific angles $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)$ through an objective that rewards alignment and penalizes directional deviation. Appendix E establishes convergence of the Pareto-stationarity residual for the deterministic RGB update under the smoothness and stepsize conditions in Assumption A.2 and the quantitative common descent condition in Assumption E.1. The toy experiment in Appendix N examines the empirical optimization trajectories.

3 METHODOLOGY

3.1 ROTATION-BASED GRADIENT BALANCING

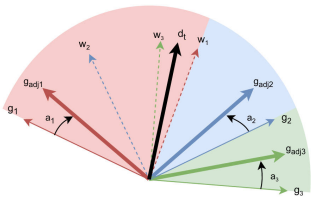


Figure 2: Methodology illustration of Rotation-based Gradient Balancing

Figure 2 illustrates the overview of RGB, where all gradients with the notation g_i are regarded as normalized gradient $\bar{g}_i$, where w_i is a state that is orthogonal to $\bar{g}_i$ and aligned with the reference direction d_t. RGB balances agreement among task gradients with fidelity to their original directions. Guided by MER principle (Section 2.3), it jointly optimizes the ideal rotation angles $\{\tilde{\alpha}_i\}_{i=1}^T$ and aggregates the resulting task vectors into a shared update.

Gradient normalization and EMA reference. Let $\theta \in \mathbb{R}^D$ denote the shared parameters and $L_1(\theta), \ldots, L_T(\theta)$ the task losses, using L_i for the same loss denoted by f_i above. We compute the gradient $g_i = \nabla_\theta L_i(\theta)$ for each task i and normalize it as $\bar{g}_i = \frac{g_i}{\|g_i\|+e}$, where $e > 0$ is a numerical stabilizer. This normalization reduces the influence of gradient magnitudes on the pairwise similarities used to optimize the rotation parameters.

Afterward, the mean direction of the gradient is computed as $y_t = \frac{1}{T}\sum_{i=1}^T \bar{g}_i$, where the cumulative y_t is adopted to compute the exponentially moving average (EMA), which serves as the reference direction for the gradient adjustment. The moving average of the gradient direction is updated as $d_t = \mu d_{t-1} + (1-\mu)\frac{y_t}{\|y_t\|+e}$. The EMA smooths the reference through an exponentially weighted history of mean directions. Proposition F.2 establishes this representation, boundedness, and convergence when the input mean direction converges. We use $\mu = 0.9$, as analyzed in the ablation study in Appendix O.

Thereafter, it remain essential to define the feasible region for gradient adjustment, in which the ideal rotation angle $\{\tilde{\alpha}_i\}_{i=1}^T$ can be optimized within a finite number of optimizations. Intuitively, the adjustment region is bound from angular of $\angle 0°$ to $90°$ from $\bar{g}_i$, where the range initial from $\bar{g}_i$ to the state w_i closest to the reference direction d_t. We define the normalized vector orthogonal to their normalized gradient $\bar{g}_i$ as $w_i = \frac{d_t - (\bar{g}_i^\top d_t)\bar{g}_i}{\|d_t - (\bar{g}_i^\top d_t)\bar{g}_i\|+e}$. Proposition F.1 gives the corresponding identities for the stabilized construction.

Rotation operator. After obtaining w_i and $\bar{g}_i$, the rotation operator is defined to rotate $\bar{g}_i$ toward the EMA direction d_t by optimizing $\alpha_i \in [0, \pi/2]$ from the objective Eq. 5, yielding $g_{\text{adj}} = r_i(\alpha_i) = \cos\alpha_i\, \bar{g}_i + \sin\alpha_i\, w_i$, where w_i is orthogonal to $\bar{g}_i$ and aligned with the projection of d_t onto the orthogonal complement of $\bar{g}_i$. This construction ensures that $r_i(\alpha_i)$ interpolates smoothly between the normalized gradient $\bar{g}_i$ (when $\alpha_i = 0$) and the orthogonal component w_i (when $\alpha_i = \frac{\pi}{2}$). Crucially, w_i serves as the reference to define the solution space for the rotation, where selecting the orthogonal vector nearest to d_t ensures the gradient rotates toward the reference direction d_t within the plane defined by $\bar{g}_i$ and w_i. For an exactly orthonormal basis $(\bar{g}_i, w_i)$, it is a unit-norm rotation and satisfies $\langle r_i(\alpha_i), \bar{g}_i \rangle = \cos\alpha_i$. The angular domain then retains nonnegative alignment with the original task direction. The alignment–fidelity condition in Theorem D.2 characterizes common descent for the final aggregated update.

Kernel-based optimization. We write $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)$ for the task-specific angle vector and $\tilde{\boldsymbol{\alpha}} = (\tilde{\alpha}_1, \ldots, \tilde{\alpha}_T)$ for the angles returned by the finite inner optimization. To avoid the computational overhead of storing and manipulating high-dimensional rotated gradients during optimization, we employ a kernel-based approach. We precompute the pairwise similarity matrices $A, B, C, E \in \mathbb{R}^{T\times T}$ where $A_{ij} = \bar{g}_i^\top \bar{g}_j$, $B_{ij} = \bar{g}_i^\top w_j$, $C_{ij} = w_i^\top \bar{g}_j$, $E_{ij} = w_i^\top w_j$. As derived in Appendix H, the pairwise similarity matrix $\mathbf{S}$ for the rotated gradients can be computed efficiently as:

$$\mathbf{S} = (\mathbf{c}\mathbf{c}^\top) \odot A + (\mathbf{c}\mathbf{s}^\top) \odot B + (\mathbf{s}\mathbf{c}^\top) \odot C + (\mathbf{s}\mathbf{s}^\top) \odot E$$

where $\mathbf{c} = \cos\boldsymbol{\alpha}$ and $\mathbf{s} = \sin\boldsymbol{\alpha}$. This allows us to optimize the rotation angles $\boldsymbol{\alpha}$ entirely in $\mathbb{R}^T$ space, making the method independent of the parameter dimension D during the iterative update. (See Appendix H for the full derivation and proof of mathematical equivalence to the original geometric formulation. See Appendix Q for comparison between original and kernel-based RGB.)

Objective for rotation angles. Subsequently, our main equation is developed to optimize the rotation angles $\alpha_1, \ldots, \alpha_T$ by minimizing a loss function that combines two terms.

$$\mathcal{L}(\boldsymbol{\alpha}) = \underbrace{\frac{1}{T(T-1)} \sum_{i<j} [1 - r_i(\alpha_i)^\top r_j(\alpha_j)]}_{\text{conflict term}} + \underbrace{\frac{1}{4T} \sum_{i=1}^{T} \left\| r_i(\alpha_i) - \bar{g}_i \right\|^2}_{\text{fidelity term}}. \tag{5}$$

The conflict term encourages mutual alignment among rotated gradients by minimizing the average pairwise angular discrepancy, scaled to the $[0, 1]$ interval via the factor $\frac{1}{2}$ (since the original range of $1 - \cos\theta$ is $[0, 2]$). Under exact unit normalization, it has the equivalent form

$$\frac{1}{T(T-1)} \sum_{i<j} (1 - r_i^\top r_j) = \frac{T}{2(T-1)} \left(1 - \left\| \frac{1}{T} \sum_{i=1}^{T} r_i \right\|^2 \right) = \frac{T}{2(T-1)} \left(1 - \|v\|^2 \right),$$

so minimizing it increases the coherence of the mean rotated vector. Similarly, the fidelity term penalizes deviation from their normalized task gradient $\bar{g}_i$, and is normalized to the $[0, 1]$ range using a factor of $\frac{1}{4}$ (as the squared distance between two opposite unit vectors reaches a maximum of 4). The normalization ensures that both terms are balanced on a comparable numerical scale, preventing either from dominating the loss due to differences in magnitude and promoting fair contribution to the objective. Since both r_i and $\bar{g}_i$ are unit vectors, it can be simplified to

$$\frac{1}{4T} \sum_{i=1}^{T} \|r_i(\alpha_i) - \bar{g}_i\|^2 = \frac{1}{2T} \sum_{i=1}^{T} (1 - \cos\alpha_i).$$

Its marginal penalty, $\sin\alpha_i/(2T)$, increases over the admissible interval, providing increasing resistance to larger rotations. Together, these terms implement an alignment–fidelity trade-off in the task-gradient geometry. Proposition G.6 in Appendix G.3 establishes a quantitative connection between mutual alignment among transformed gradients and their cross-task projections onto the original normalized task gradients. Under the stated assumptions, the discrepancy is bounded by twice the square root of the fidelity term. Thus, the fidelity penalty controls the discrepancy between the alignment quantity optimized by RGB and the normalized cross-task projections underlying its first-order motivation in Eq. 1. Theorem C.1 establishes the existence of a global minimizer of the rotation objective $\mathcal{L}$.

Algorithm 1 Rotation-Based Gradient Balancing

Inputs: task losses $L_1, \dots, L_T$, hyper-parameters $(\mu, e, \alpha_{\text{steps}})$
State: $\theta \in \mathbb{R}^D$, EMA direction d_{t-1}
Compute task gradients $g_i = \nabla_\theta L_i(\theta)$ and normalize gradients $\bar{g}_i = g_i/(\|g_i\| + e)$
Compute mean direction $y_t = \frac{1}{T}\sum_{i=1}^{T} \bar{g}_i$
Update EMA direction $d_t = \mu d_{t-1} + (1-\mu) y_t/(\|y_t\| + e)$
Compute feasible orthogonal corrections state $w_i = \frac{d_t - (\bar{g}_i^\top d_t)\bar{g}_i}{\|d_t - (\bar{g}_i^\top d_t)\bar{g}_i\| + e}$
Compute kernels: $A_{ij} = \bar{g}_i^\top \bar{g}_j, \quad B_{ij} = \bar{g}_i^\top w_j, \quad C_{ij} = w_i^\top \bar{g}_j, \quad E_{ij} = w_i^\top w_j$
Initialize $\boldsymbol{\alpha} = \mathbf{0}$ and optimize Eq. 5 for α_{steps} iterations over $\boldsymbol{\alpha} \in [0, \pi/2]^T$, obtaining $\tilde{\boldsymbol{\alpha}}$
Set final rotated gradients $r_i \leftarrow r_i(\tilde{\alpha}_i)/(\|r_i(\tilde{\alpha}_i)\| + e)$
Compute shared update direction $v = \frac{1}{T}\sum_{i=1}^{T} r_i, \quad$ update $\theta \leftarrow \theta - \eta_t\, v$

Complete algorithm. Algorithm 1 summarizes the stabilized numerical procedure. It evaluates the conflict term using the precomputed kernels and uses the angular fidelity expression $\frac{1}{2T}\sum_i(1 - \cos\alpha_i)$ as a numerical approximation to the squared-distance term. After obtaining $\tilde{\boldsymbol{\alpha}}$, we apply the rotation operator, normalize the adjusted vectors, and average them to form a shared descent direction v. The shared parameters are updated by $\theta \leftarrow \theta - \eta_t v$. Under the smoothness and stepsize conditions in Assumption A.2 and the quantitative common descent condition in Assumption E.1, the deterministic update satisfies $\rho(\theta_t) \to 0$, and every accumulation point is Pareto stationary (Theorem E.2).

4 Results and Discussion

4.1 Evaluation Metrics

We evaluate our proposed method on several multi-task learning benchmarks, including NYUv2 Silberman et al. (2012), Cityscapes Cordts et al. (2016), QM9 Blum & Reymond (2009); Ramakrishnan et al. (2014), and CelebA Liu et al. (2015). To assess performance, we follow standard practices in multi-task learning from Nash-MTL Navon et al. (2022) and FAMO Liu et al. (2023). To ensure a fair comparison, we do not naively aggregate baseline metrics from disparate original publications. Instead, our evaluation protocol aligns each dataset with a specific, community-standardized benchmark codebase and its correspondingly validated metrics. For each task i, we compute the per-task performance drop $\Delta m\%$ relative to the STL baseline: $\Delta m\% = \frac{1}{K}\sum_{k=1}^{K}(-1)^{\delta_k}\frac{M_{m,k} - M_{b,k}}{M_{b,k}} \times 100$, where K is the number of total metrics, $M_{b,k}$ is the STL baseline value for metric M_k, $M_{m,k}$ is the value from the MTL method, and $\delta_j = 1$ if a higher value is better for a metric M_k (and 0 otherwise). A lower (more negative) $\Delta m\%$ indicates better performance relative to STL. We also report the mean rank (MR) across metrics or tasks, where lower MR signifies superior overall ranking. MR $= \frac{1}{T}\sum_{i=1}^{T}\frac{1}{M_i}\sum_{j=1}^{M_i} R_{i,j}$ where M_i is the number of metrics for task i, and $R_{i,j}$ is the rank of the method on metric j of task i (with rank 1 representing the best performance among compared methods).

4.2 Results on NYUv2, QM9, Cityscapes and CelebA

The NYUv2 dataset comprises 1,449 densely annotated RGB-D images of indoor scenes, focusing on three pixel-level tasks: 13-class semantic segmentation (measured by mean Intersection over Union (mIoU) and pixel accuracy), monocular depth estimation (absolute and relative error), and surface normal prediction (mean and median angle distances, and percentages of pixels within 11.25°, 22.5°, and 30° thresholds). For NYUv2, we follow the public LibMTL Lin & Zhang (2023) implementation to ensure consistency of comparison against the baseline metrics reported in ConsMTL Qin et al. (2025b). We use a SegNet architecture with a Multi-Task Attention Network (MTAN) Liu et al. (2019). The model is trained for 200 epochs with a batch size of 2. We optimize the network using Adam with an initial learning rate of 1e-4 and zero weight decay, applying a step decay of 0.5 every 100 epochs. Consistent with the original pipeline, the training data is processed with data augmentation enabled. As shown in Table 1, RGB ($\alpha_{\text{steps}} = 10$) achieves a $\Delta m\%$ of -8.61

compared to PIVRG (-6.50) and ConsMTL (-6.72). Notably, RGB excels in segmentation and depth metrics, with a low MR confirming stable performance across all tasks. In addition, we also include the comparison between RotoGrad Javaloy & Valera (2022) in NYUv2 dataset. Since RotoGrad uses a different architectural setup for SegNet, a separate comparison is detailed in Appendix P, Table 11.

Table 1: Results on NYUv2. MR values are recomputed using all methods. Lower is better for ↓, higher for ↑.

Method	Segmentation		Depth		Surface Normal					MR ↓	$\Delta m\%$ ↓
					Angle Dist. ↓		Within t° ↑				
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Mean	Median	11.25	22.5	30		
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15		
LS Kendall et al. (2018)	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	16.33	5.59
SI Ruder (2017)	38.45	64.27	0.5354	0.2201	27.60	23.37	22.53	48.57	62.32	14.67	4.39
RLW Lin et al. (2022)	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	19.22	7.78
DWA Liu et al. (2019)	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	15.33	3.57
UW Kendall et al. (2018)	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	15.00	4.05
MGDA-UB Sener & Koltun (2018)	30.47	59.90	0.6070	0.2555	24.88	19.45	29.18	56.88	69.36	12.56	1.38
PCGrad Yu et al. (2020)	38.06	64.64	0.5550	0.2325	27.41	22.80	23.86	49.83	63.14	15.67	3.97
GradDrop Chen et al. (2020)	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	14.44	3.58
CAGrad Liu et al. (2021a)	39.79	65.49	0.5486	0.2250	26.31	21.58	25.61	52.36	65.58	11.00	0.20
IMTL-G Liu et al. (2021b)	39.35	65.60	0.5426	0.2256	26.02	21.19	26.20	53.13	66.24	10.33	-0.76
MoCo Fernando et al. (2023)	40.30	66.07	0.5575	0.2135	26.67	21.83	25.61	51.78	64.85	10.22	0.16
Nash-MTL Navon et al. (2022)	40.13	65.93	0.5261	0.2171	25.26	20.08	28.40	55.47	68.15	7.56	-4.04
FAMO Liu et al. (2023)	38.88	64.90	0.5474	0.2194	25.06	19.57	29.21	56.61	68.98	9.11	-4.10
FairGrad Ban & Ji (2024)	39.74	66.01	0.5377	0.2236	24.84	19.60	29.26	56.58	69.16	7.00	-4.66
BiLB4MTL (τ=1) Xiao et al. (2025)	38.04	63.90	0.5402	0.2278	24.70	19.19	29.97	57.44	69.69	8.89	-4.40
Aligned-MTL Senushkin et al. (2023)	40.82	66.33	0.5300	0.2200	25.19	19.71	28.88	56.23	68.54	6.56	-4.93
GO4Align Shen et al. (2024)	40.42	65.37	0.5492	0.2167	24.76	18.94	30.54	57.87	69.84	5.44	-6.08
PIVRG Qin et al. (2025a)	39.90	65.74	0.5365	0.2243	**24.30**	18.80	30.95	58.26	**70.38**	3.89	-6.50
ConsMTL Qin et al. (2025b)	40.33	65.32	0.5491	0.2151	24.35	18.80	**31.06**	**58.28**	70.31	4.22	-6.72
RGB	**44.25**	**68.84**	**0.5125**	**0.2126**	24.47	**18.80**	30.07	57.70	70.12	**2.11**	**-8.61**

The Cityscapes and the CelebA datasets are two widely used multi-task learning benchmarks. Cityscapes consists of 5,000 high-quality, pixel-level annotated images of urban street scenes collected from 50 cities, and includes three tasks: semantic segmentation (mIoU), disparity estimation (L1 pixel error), and instance segmentation (MSE). CelebA is a large-scale facial attributes dataset containing over 200,000 celebrity images, each annotated with 40 binary attributes, which can be formulated as 40 binary classification tasks. Together, these datasets provide diverse and challenging scenarios for evaluating multi-task learning methods, spanning both complex scene understanding and fine-grained facial attribute recognition.

We base all experiments on publicly available implementations to ensure fair comparisons with prior work. For the Cityscapes benchmark, we use the Align-MTL Senushkin et al. (2023) codebase, and our results are comparing against the baseline metrics reported in Senushkin et al. (2023). This includes a PSPNet architecture with a dilated ResNet-50 encoder, trained for 60 epochs using the Adam optimizer with a base learning rate of 1e-4. Batch sizes are 8 for training and 4 for testing. We adopt the default learning rate schedule, decaying the rate by a factor of 0.7 every 3 epochs after an initial 40-epoch milestone. Training data undergoes standard augmentations, including random horizontal flipping and random rotations up to 10 degrees. For CelebA, we build upon the FAMO Liu et al. (2023) codebase, comparing against the standardized baseline metrics validated by ConsMTL Qin et al. (2025b). We employ a shallow 9-layer convolutional neural network, similar to the architecture used in Chen et al. (2020); Liu et al. (2023); Ban & Ji (2024), followed by task-specific fully-connected prediction layers. Input images are resized to 64×64. The model is trained for 15 epochs using Adam, with a batch size of 256 and a constant learning rate of 3e-4.

As shown in Table 2, RGB configured with a α_{steps} of 30 achieves a $\Delta m\%$ of -1.70 on Cityscapes. Notably, the MR of 3.00 indicates that our method not only improves absolute task performance but also consistently achieves a higher relative ranking across all tasks when compared to prior methods. On the CelebA benchmark, which involves 40 concurrent facial attribute classification tasks, RGB ($\alpha_{\text{steps}} = 100$) achieves the best overall $\Delta m\%$ of -1.49, while also maintaining the lowest MR of 3.23 among the compared methods.

Table 2: Results on Cityscapes (3 tasks), CelebA (40 tasks), and QM9 (11 tasks). MR is recomputed as the mean rank across the task metrics. Lower is better unless otherwise indicated.

	Cityscapes					CelebA			QM9		
Method	mIoU [%] ↑	L1 [px] ↓	MSE ↓	MR ↓	$\Delta m\%$ ↓	Method	MR ↓	$\Delta m\%$ ↓	Method	MR ↓	$\Delta m\%$ ↓
STL	66.73	10.55	0.33			STL			STL		
Baseline:Uniform	52.98	10.89	0.39	12.67	14.30	LS	9.35	4.15	LS	11.45	177.6
RLW	51.26	10.25	0.41	13.00	15.58	SI	11.32	7.20	SI	8.55	77.8
DWA	53.15	10.22	0.40	10.67	13.20	RLW	8.30	1.46	RLW	13.00	203.8
Uncertainty	60.12	**9.87**	0.33	5.00	1.53	DWA	10.10	3.20	DWA	11.18	175.3
MGDA	66.72	17.02	0.33	7.33	20.62	UW	8.57	3.23	UW	10.09	108.0
MGDA-UB	66.37	18.63	0.32	6.67	25.05	MGDA-UB	14.88	14.85	MGDA-UB	11.64	120.5
GradNorm	57.24	10.29	0.35	9.00	6.55	PCGrad	9.78	3.17	PCGrad	9.55	125.7
GradDrop	52.98	10.09	0.40	10.33	12.50	GradDrop	11.30	3.29	CAGrad	10.73	112.8
PCGrad	54.06	9.91	0.38	8.33	10.00	CAGrad	9.35	2.48	IMTL-G	9.27	77.2
GradVac	54.07	10.39	0.40	11.00	12.89	IMTL-G	7.40	0.84	Nash-MTL	6.64	62.0
CAGrad	64.33	10.15	0.34	7.00	1.46	Nash-MTL	7.88	2.84	FAMO	8.09	58.5
IMTL	65.13	11.58	0.32	6.67	3.10	FAMO	7.60	1.21	FairGrad	7.00	57.9
Nash-MTL	64.84	11.90	0.37	10.33	9.38	FairGrad	8.20	0.37	ConsMTL	3.27	23.2
Align-MTL	**67.06**	10.63	0.33	5.67	-0.02	PIVRG	4.33	-0.96	GO4Align	7.73	52.7
Align-MTL-UB	66.07	10.54	0.32	5.00	-0.35	ConsMTL	4.42	-1.42	BiLB4MTL ($\tau = \sigma$)	5.82	49.5
RGB	65.72	10.06	**0.32**	**3.00**	**-1.70**	RGB	**3.23**	**-1.49**	RGB	**1.64**	**12.4**

The QM9 dataset is a quantum chemistry benchmark comprising about 130,000 stable organic molecules represented as graphs and associated with 11 quantum chemical properties (geometric, energetic, electronic, and thermodynamic). Our QM9 experiment is implemented using the Nash-MTL Navon et al. (2022) codebase architecture, mirroring the exact evaluation setup and corresponding baseline metrics reported in ConsMTL Qin et al. (2025b). Following prior works Navon et al. (2022); Liu et al. (2023), we adopt the neural message passing network (MPNN) architecture introduced by Gilmer et al. (2017), with a sequence of message passing layers and a Set2Set pooling operator Vinyals et al. (2015). Using the standard PyTorch Geometric implementation Fey & Lenssen (2019), we split the dataset into 110K molecules for training, 10K for validation, and 10K for testing. The models are trained for 300 epochs using a standard learning rate scheduler, consistent with prior works. Table 2 demonstrates that our RGB method achieves the lowest $\Delta m\%$ (12.4) and MR (1.64), outperforming ConsMTL (23.2) by a substantial numerical margin. On this benchmark, our RGB method was evaluated with a fixed α_{steps} of 4 as its hyperparameters. Notably, RGB outperformed single-task learning (STL) on 5 out of the 11 regression tasks in Table 6, showing its ability to leverage cross-task information even under the complex multi-task setting. The detailed per-seed standard errors for all datasets are reported in Appendix P (Tables 7–10).

5 Conclusion and Limitations

In this work, we introduce Rotation-Based Gradient Balancing (RGB), a gradient manipulation method for Multi-Task Learning (MTL) that jointly optimizes task-gradient rotations to balance alignment with fidelity to the original gradient directions. Our empirical analysis supports the alignment–fidelity trade-off underlying the Multi-Task Equilibrium Relationship (MER): increasing the degree of gradient modification does not consistently improve task performance, highlighting the importance of controlling rotation magnitude. Across the evaluated benchmarks, RGB achieves competitive performance against recent methods, including PIVRG and ConsMTL. Ablation studies on the sensitivity of inner-iteration budget, α_{steps} (J), the EMA direction parameter (O), tolerance limits (K), MER distances and performance analysis (L), and gradient miss-correction (M) provide additional empirical support for these concepts across multiple datasets. Importantly, these performance gains do not incur prohibitive computational costs. As detailed in Appendix R, our kernel-based formulation decouples the inner optimization from the parameter dimension D, resulting in a time complexity comparable to standard projection methods like PCGrad. One limitation of RGB is its sensitivity to the hyperparameter α_{steps}, which controls the update steps of the inner optimization loop. The appropriate α_{steps} depends on the dataset and training dynamics, and should therefore be interpreted as tuned operating points rather than a universal default. In practice, we recommend starting at $\alpha_{\text{steps}} = 5$ and evaluating progressively larger budgets from a small candidate set such as $\{5, 10, 30, 50\}$. We leave the investigation of stopping criteria that determine the inner optimization budget for future work. The code will be made publicly available.

AI USE STATEMENT

In this work, we used generative AI tools to check mathematical notation alignment, correct grammar, and rephrase sentences. We did not use generative AI tools for ideation, methodology design, experimental execution, or result analysis. Additionally, we used generative AI tools as simulated reviewers to provide critical feedback on our manuscript. We have carefully reviewed all AI-assisted work to ensure content validity and checked the LLM-generated text to avoid potential plagiarism. All LLM-generated content was verified and tested for correctness by two authors. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We confirm that this work presents no ethical concerns. Our research utilizes publicly available standard datasets and benchmarks for multi-task learning. All experimental procedures, model designs, and analyses comply with standard research integrity guidelines, and we have no conflicts of interest to disclose.

REPRODUCIBILITY STATEMENT

To ensure the full reproducibility of our results, we provide comprehensive details regarding our experimental setup, hyperparameter configurations, and architectural designs within the main paper and appendix. Specifically, all models are implemented using standard public datasets, and the detailed setup can be found in Section 4. Furthermore, the complete source code, training scripts, evaluation pipelines, and supplementary experiments will be made publicly available upon publication to facilitate independent replication.

REFERENCES

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 2715–2731, 2024.

Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25): 8732–8733, 2009.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

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

Heshan Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *International Conference on Learning Representations*, 2023.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. Pmlr, 2017.

Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

Baijiong Lin and Yu Zhang. LibMTL: A Python library for multi-task learning. *Journal of Machine Learning Research*, 24(209):1–7, 2023.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

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

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1871–1880, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16428–16446, 2022.

Xiaohan Qin, Xiaoxing Wang, and Junchi Yan. Revisiting fairness in multitask learning: A performance-driven approach for variance reduction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20492–20501, 2025a.

Xiaohan Qin, Xiaoxing Wang, and Junchi Yan. Towards consistent multi-task learning: Unlocking the potential of task-specific parameters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10067–10076, 2025b.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

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

Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20083–20093, 2023.

Jiayi Shen, Qi Wang, Zehao Xiao, Nanne Van Noord, and Marcel Worring. Go4align: Group optimization for multi-task alignment. *Advances in Neural Information Processing Systems*, 37: 111382–111405, 2024.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pp. 746–760. Springer, 2012.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations*, 2015.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021.

Peiyao Xiao, Chaosheng Dong, Shaofeng Zou, and Kaiyi Ji. Scalable bilevel loss balancing for multi-task learning. *arXiv e-prints*, pp. arXiv–2502, 2025.

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.

Zhi Zhang, Jiayi Shen, Congfeng Cao, Gaole Dai, Shiji Zhou, Qizhe Zhang, Shanghang Zhang, and Ekaterina Shutova. Proactive gradient conflict mitigation in multi-task learning: A sparse training perspective. *arXiv preprint arXiv:2411.18615*, 2024.

A PRELIMINARIES

As in Section 2, we consider unconstrained shared parameters $\Theta = \mathbb{R}^D$ and differentiable task losses $f_i : \mathbb{R}^D \to \mathbb{R}$, $i = 1, \dots, T$, with $T \geq 2$, and write $g_i = \nabla f_i(\theta)$. Task indices are denoted by i, j, and outer iterations by t. The vector v is used in the update $\theta - \eta v$; its negative $-v$ is a common descent direction when $\langle g_i, v \rangle > 0$ for every task. The vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_T)$ contains the task-specific rotation parameters. We denote a global minimizer of the inner objective by $\boldsymbol{\alpha}^\star$ and the angles returned by the finite inner optimization by $\tilde{\boldsymbol{\alpha}} = (\tilde{\alpha}_1, \dots, \tilde{\alpha}_T)$.

Definition A.1 (Lipschitz-Continuous Gradient)**.** A differentiable function f has a β-Lipschitz continuous gradient if $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$ for all $x, y \in \mathbb{R}^D$.

Assumption A.2 (Smoothness and Stepsizes)**.** The task losses and outer stepsizes satisfy:

- **(A1)** Each task loss is bounded from below: $\inf_\theta f_i(\theta) > -\infty$.
- **(A2)** Each task loss has a β-Lipschitz continuous gradient, for a common constant $\beta > 0$.
- **(A3)** The stepsizes satisfy $\eta_t > 0$, $\sum_{t=0}^{\infty} \eta_t = \infty$, and $\sum_{t=0}^{\infty} \eta_t^2 < \infty$.

The stepsize conditions provide an unbounded cumulative stepsize sum and a summable smoothness remainder for bounded update directions.

Definition A.3 (Pareto Optimality and Stationarity)**.** A point $\theta^\star$ is Pareto optimal if there is no θ such that $f_i(\theta) \leq f_i(\theta^\star)$ for every i, with a strict inequality for at least one task. It is weakly Pareto optimal if there is no θ such that $f_i(\theta) < f_i(\theta^\star)$ for every i. A point is Pareto stationary if

$$0 \in \operatorname{conv}\{\nabla f_1(\theta), \dots, \nabla f_T(\theta)\}.$$

To quantify first-order stationarity, define

$$\rho(\theta) := \min_{\lambda \in \Delta_T} \left\| \sum_{i=1}^{T} \lambda_i \nabla f_i(\theta) \right\|, \qquad \Delta_T := \left\{ \lambda \in \mathbb{R}^T : \lambda_i \geq 0,\ \sum_{i=1}^{T} \lambda_i = 1 \right\}. \tag{6}$$

Thus $\rho(\theta) = 0$ exactly at Pareto stationary points. For unconstrained parameters, separation of the gradient convex hull from the origin shows that this characterization is equivalent to the absence of a vector having strictly negative inner products with every task gradient, as in Section 2.1. This is the minimum-norm convex-hull characterization used in multi-gradient descent Désidéri (2012).

B A FUNDAMENTAL GEOMETRIC BOUND

Lemma B.1 (Necessary Pairwise-Cosine Bound)**.** *Let $u_1, \dots, u_T \in \mathbb{R}^D$ be unit vectors satisfying $\sum_{i=1}^{T} u_i = 0$. Then*

$$\max_{i \neq j} (-u_i^\top u_j) \geq \frac{1}{T-1}.$$

Proof. Expanding the squared norm of the sum gives

$$0 = \left\| \sum_{i=1}^{T} u_i \right\|^2 = T + \sum_{i \neq j} u_i^\top u_j.$$

The average of the $T(T-1)$ off-diagonal inner products is therefore $-1/(T-1)$, so at least one is no greater than this average. □

Lemma B.2 (Sufficient Condition for Nonnegative Mean-Alignment)**.** *Let $v_1, \dots, v_T \in \mathbb{R}^D$ be unit vectors and set $\bar{v} = \frac{1}{T} \sum_{k=1}^{T} v_k$. If $v_i^\top v_j \geq -1/(T-1)$ for every $i \neq j$, then $\langle v_i, \bar{v} \rangle \geq 0$ for every i. More generally, if*

$$v_i^\top v_j \geq -\frac{1}{T-1} + \delta \qquad (i \neq j)$$

for some $\delta > 0$, then

$$\langle v_i, \bar{v} \rangle \geq \frac{T-1}{T} \delta > 0 \qquad (i = 1, \dots, T).$$

Proof. For each i,

$$\langle v_i, \bar{v} \rangle = \frac{1}{T} \left(1 + \sum_{j \neq i} v_i^\top v_j \right).$$

Substitution of the respective pairwise bounds proves both claims. □

C Existence of Optimal Rotation Angles

Fix the task-gradient and consensus directions at an outer iteration, with the rotation construction well defined. The inner objective $\mathcal{L}$ is the conflict–fidelity objective defined in the main text.

Theorem C.1 (Existence of a Minimiser)**.** *If each $r_i(\alpha_i)$ is continuous in α_i, then $\mathcal{L}$ attains a global minimum on $[0, \pi/2]^T$. In particular, there exists $\boldsymbol{\alpha}^\star \in [0, \pi/2]^T$ such that*

$$\mathcal{L}(\boldsymbol{\alpha}^\star) = \min_{\boldsymbol{\alpha} \in [0, \pi/2]^T} \mathcal{L}(\boldsymbol{\alpha}).$$

Proof. The inner objective consists of finite sums of inner products and squared norms of continuous rotation mappings, and is therefore continuous. The feasible set $[0, \pi/2]^T$ is compact. The result follows from the Weierstrass extreme value theorem. □

Proposition C.2 (Pareto Efficiency of the Inner Trade-off)**.** *Every global minimizer $\boldsymbol{\alpha}^\star$ of the RGB inner objective is Pareto optimal with respect to its conflict and fidelity terms over the feasible angular domain.*

Proof. Both terms enter $\mathcal{L}$ with strictly positive coefficients. If another feasible angle vector weakly decreased both terms and strictly decreased at least one, it would strictly decrease $\mathcal{L}$, contradicting global optimality of $\boldsymbol{\alpha}^\star$. □

D Common Descent Through Alignment and Fidelity

Definition D.1 (Rotation-Adjusted Direction)**.** For a feasible angle vector $\boldsymbol{\alpha} \in [0, \pi/2]^T$, define

$$r_i := r_i(\alpha_i), \qquad v := \frac{1}{T} \sum_{i=1}^{T} r_i.$$

For each nonzero task gradient, write $\bar{g}_i = g_i / \|g_i\|$.

Theorem D.2 (Alignment–Fidelity Descent Bound)**.** *Suppose $g_i \neq 0$ for every i. For any feasible angle vector,*

$$\langle g_i, v \rangle \geq \|g_i\| \left(\langle r_i, v \rangle - \|r_i - \bar{g}_i\| \, \|v\| \right). \tag{7}$$

Consequently, if

$$\langle r_i, v \rangle > \|r_i - \bar{g}_i\| \, \|v\| \qquad (i = 1, \ldots, T), \tag{8}$$

*then $v \neq 0$ and $-v$ is a strict common descent direction for all task losses. Under Assumption **(A2)**, every stepsize satisfying*

$$0 < \eta < \min_i \frac{2 \langle g_i, v \rangle}{\beta \|v\|^2}$$

gives $f_i(\theta - \eta v) < f_i(\theta)$ for every task.

Proof. Using $g_i = \|g_i\| \bar{g}_i$ and the Cauchy–Schwarz inequality,

$$\begin{aligned} \langle g_i, v \rangle &= \|g_i\| \big(\langle r_i, v \rangle + \langle \bar{g}_i - r_i, v \rangle \big) \\ &\geq \|g_i\| \big(\langle r_i, v \rangle - \|r_i - \bar{g}_i\| \, \|v\| \big). \end{aligned}$$

Condition Eq. 8 makes this expression strictly positive for every task and also excludes $v = 0$. The smoothness inequality gives

$$f_i(\theta - \eta v) \leq f_i(\theta) - \eta \langle g_i, v \rangle + \frac{\beta}{2} \eta^2 \|v\|^2,$$

which is strictly smaller than $f_i(\theta)$ for the stated stepsizes. □

For the unit-rotation construction $r_i = \cos\alpha_i \bar{g}_i + \sin\alpha_i w_i$, with $\|w_i\| = \|\bar{g}_i\| = 1$ and $w_i^\top \bar{g}_i = 0$,

$$\|r_i - \bar{g}_i\| = 2\sin(\alpha_i/2).$$

Hence the descent certificate compares alignment with the averaged direction against the directional deviation induced by rotation. In particular, if the rotated unit vectors satisfy $r_i^\top r_j \geq -1/(T-1)+\delta$ for every $i \neq j$, then Lemma B.2 yields the sufficient condition

$$\frac{T-1}{T}\delta > 2\sin(\alpha_i/2)\|v\| \qquad (i = 1, \ldots, T).$$

This relates the pairwise alignment margin directly to the admissible rotation-induced deviation.

E Convergence Under a Quantitative Descent Condition

We analyze the deterministic outer update

$$\theta_{t+1} = \theta_t - \eta_t v_t, \qquad v_t = \frac{1}{T}\sum_{i=1}^{T} r_i(\tilde{\alpha}_{i,t}), \tag{9}$$

where $g_{i,t} = \nabla f_i(\theta_t)$ are exact task gradients and $\tilde{\boldsymbol{\alpha}}_t = (\tilde{\alpha}_{1,t}, \ldots, \tilde{\alpha}_{T,t})$ denotes the angles returned by the inner optimization. The rotation construction is assumed to be well defined along the sequence, with $\|r_i(\tilde{\alpha}_{i,t})\| \leq 1$.

Assumption E.1 (Quantitative Common Descent)**.** There exists a constant $\kappa > 0$, independent of t, such that

$$\langle g_{i,t}, v_t\rangle \geq \kappa\rho(\theta_t)^2 \qquad \text{for every } i \text{ and } t.$$

For nonzero task gradients, Theorem D.2 provides the following sufficient geometric condition for Assumption E.1:

$$\|g_{i,t}\|\left(\langle r_i(\tilde{\alpha}_{i,t}), v_t\rangle - \|r_i(\tilde{\alpha}_{i,t}) - \bar{g}_{i,t}\|\,\|v_t\|\right) \geq \kappa\rho(\theta_t)^2.$$

Theorem E.2 (Convergence to Pareto Stationarity)**.** *Under Assumptions A.2 and E.1, the iterates in Eq. 9 satisfy*

$$\lim_{t\to\infty} \rho(\theta_t) = 0.$$

Every accumulation point of the sequence is Pareto stationary. If the sequence is bounded, at least one such accumulation point exists. Boundedness holds, in particular, if $\Phi(\theta) = \sum_i f_i(\theta)$ is coercive.

Proof. Set $\Phi(\theta) = \sum_{i=1}^{T} f_i(\theta)$ and $\Phi_{\inf} = \sum_{i=1}^{T} \inf_\theta f_i(\theta) > -\infty$. Since $\|v_t\| \leq 1$, smoothness and Assumption E.1 imply

$$\begin{aligned}\Phi(\theta_{t+1}) &\leq \Phi(\theta_t) - \eta_t \sum_{i=1}^{T}\langle g_{i,t}, v_t\rangle + \frac{\beta T}{2}\eta_t^2\|v_t\|^2 \\ &\leq \Phi(\theta_t) - \kappa T\eta_t\rho(\theta_t)^2 + \frac{\beta T}{2}\eta_t^2.\end{aligned}$$

Summing and using the lower bound gives

$$\kappa T\sum_{t=0}^{\infty}\eta_t\rho(\theta_t)^2 \leq \Phi(\theta_0) - \Phi_{\inf} + \frac{\beta T}{2}\sum_{t=0}^{\infty}\eta_t^2 < \infty. \tag{10}$$

We next establish continuity control of the residual. Let λ minimize Eq. 6 at y. Then

$$\begin{aligned}\rho(x) &\leq \left\|\sum_i \lambda_i \nabla f_i(x)\right\| \\ &\leq \rho(y) + \sum_i \lambda_i\|\nabla f_i(x) - \nabla f_i(y)\| \\ &\leq \rho(y) + \beta\|x - y\|.\end{aligned}$$

Interchanging x and y shows that ρ is β-Lipschitz. It follows that

$$|\rho(\theta_{t+1}) - \rho(\theta_t)| \le \beta\|\theta_{t+1} - \theta_t\| \le \beta\eta_t. \tag{11}$$

Write $\rho_t = \rho(\theta_t)$. Eq. 10 and $\sum_t \eta_t = \infty$ first imply $\liminf_t \rho_t = 0$. If ρ_t did not converge to zero, there would exist $\varepsilon > 0$ and infinitely many disjoint intervals $[p, q)$ with $\rho_p \ge \varepsilon$, $\rho_q \le \varepsilon/2$, and $\rho_t > \varepsilon/2$ for $p \le t < q$. By Eq. 11, each interval satisfies

$$\sum_{t=p}^{q-1} \eta_t \ge \frac{\varepsilon}{2\beta}, \qquad \sum_{t=p}^{q-1} \eta_t \rho_t^2 \ge \frac{\varepsilon^3}{8\beta}.$$

Infinitely many such intervals contradict Eq. 10. Therefore $\rho_t \to 0$. For any convergent subsequence $\theta_{t_k} \to \theta^\star$, continuity gives $\rho(\theta^\star) = 0$, proving Pareto stationarity. Boundedness ensures existence of a convergent subsequence. Finally, the aggregate-loss inequality also gives

$$\Phi(\theta_t) \le \Phi(\theta_0) + \frac{\beta T}{2} \sum_{s=0}^{\infty} \eta_s^2.$$

If Φ is coercive, this uniform upper bound places the entire sequence in a bounded sublevel set. □

Corollary E.3 (Pareto Optimality for Convex Task Losses)**.** *Suppose the assumptions of Theorem E.2 hold and each f_i is convex on $\mathbb{R}^D$. Every accumulation point $\theta^\star$ is weakly Pareto optimal. If, in addition, there exists $\lambda^\star \in \Delta_T$ with $\lambda_i^\star > 0$ for every i and*

$$\sum_{i=1}^{T} \lambda_i^\star \nabla f_i(\theta^\star) = 0,$$

then $\theta^\star$ is Pareto optimal.

Proof. Pareto stationarity supplies $\lambda^\star \in \Delta_T$ with a zero weighted gradient. By convexity, for every θ,

$$\sum_i \lambda_i^\star f_i(\theta) \ge \sum_i \lambda_i^\star f_i(\theta^\star) + \left\langle \sum_i \lambda_i^\star \nabla f_i(\theta^\star), \theta - \theta^\star \right\rangle = \sum_i \lambda_i^\star f_i(\theta^\star).$$

Strict improvement in every task would strictly decrease this weighted sum, establishing weak Pareto optimality. If all weights are strictly positive, weak improvement in every task with strict improvement in at least one would also strictly decrease the weighted sum. This proves Pareto optimality. □

F Theoretical Analysis

We analyze the geometry of the RGB transformation, the temporal smoothing induced by the EMA reference, and the first-order effect of the aggregated update on the task losses. We then derive a cost–benefit decomposition and relate the RGB objective to its alignment and fidelity components.

Proposition F.1 (Orthogonal Decomposition of the Rotation Operator)**.** *Let $\bar{g}_i \in \mathbb{R}^D$ be a unit vector and $d_t \in \mathbb{R}^D$ a reference direction. For $e > 0$, define*

$$w_i' = d_t - (\bar{g}_i^\top d_t)\bar{g}_i, \qquad w_i = \frac{w_i'}{\|w_i'\| + e}.$$

Then $\bar{g}_i^\top w_i = 0$ and

$$\|w_i\| = \frac{\|w_i'\|}{\|w_i'\| + e} \le 1.$$

For $\alpha_i \in [0, \pi/2]$, the transformation $r_i(\alpha_i) = \cos\alpha_i\, \bar{g}_i + \sin\alpha_i\, w_i$ satisfies

$$\|r_i(\alpha_i)\|^2 = \cos^2\alpha_i + \sin^2\alpha_i\, \|w_i\|^2 \le 1, \tag{12}$$

$$\bar{g}_i^\top r_i(\alpha_i) = \cos\alpha_i, \tag{13}$$

$$r_i(0) = \bar{g}_i, \qquad r_i(\pi/2) = w_i.$$

For $w_i' \neq 0$, exact normalization $w_i = w_i'/\|w_i'\|$ gives a unit-norm rotation through angle α_i.

Proof. Since $\|\bar{g}_i\| = 1$,

$$\bar{g}_i^\top w_i' = \bar{g}_i^\top d_t - (\bar{g}_i^\top d_t)\|\bar{g}_i\|^2 = 0.$$

Scaling preserves orthogonality and gives the stated norm of w_i. Expanding $\|r_i(\alpha_i)\|^2$ and using $\bar{g}_i^\top w_i = 0$ proves Eq. 12. Taking the inner product with $\bar{g}_i$ proves Eq. 13. The endpoint identities follow by substitution. Under exact normalization, $\bar{g}_i$ and w_i form an orthonormal pair, so the transformation is a unit-norm planar rotation. □

The reference d_t determines the orthogonal component toward which the transformation proceeds. The endpoint w_i is this orthogonal component, while $r_i(0)$ is the original normalized gradient. If $w_i' = 0$, the regularized construction gives $r_i(\alpha_i) = \cos\alpha_i\, \bar{g}_i$.

Proposition F.2 (EMA Representation and Stability)**.** *Let* $y_t = \frac{1}{T}\sum_{i=1}^{T} \bar{g}_i(\theta_t)$ *and, for* $t \geq 1$*, define*

$$d_t = \mu d_{t-1} + (1-\mu)\frac{y_t}{\|y_t\| + e}, \qquad 0 \leq \mu < 1, \quad e > 0.$$

Then

$$d_t = \mu^t d_0 + (1-\mu)\sum_{s=1}^{t} \mu^{t-s}\frac{y_s}{\|y_s\| + e}, \tag{14}$$

and

$$\|d_t\| \leq \mu^t\|d_0\| + 1 - \mu^t.$$

If $y_t \to y_\infty$*, then* $d_t \to y_\infty/(\|y_\infty\| + e)$*. If additionally* $y_\infty \neq 0$*, then* $d_t/\|d_t\| \to y_\infty/\|y_\infty\|$.

Proof. Unrolling the recurrence yields Eq. 14. The norm bound follows from $\|y_s/(\|y_s\| + e)\| \leq 1$ and the geometric sum $(1-\mu)\sum_{s=1}^{t}\mu^{t-s} = 1 - \mu^t$. If $y_t \to y_\infty$, continuity of $y \mapsto y/(\|y\| + e)$ gives convergence of the EMA inputs to $y_\infty/(\|y_\infty\| + e)$. In the unrolled expression for the difference from this limit, any fixed initial segment decays geometrically and the remaining tail is bounded by the largest input error in that tail. Both contributions tend to zero. For a nonzero limit, normalizing d_t then gives the stated directional convergence. □

Eq. 14 characterizes the reference as an exponentially weighted history of normalized mean-gradient directions. The initialization contribution decays at rate μ^t.

F.1 First-Order Analysis

Fix an iterate θ with nonzero task gradients $g_i = \nabla f_i(\theta)$, and write $\bar{g}_i = g_i/\|g_i\|$. Let $v = \frac{1}{T}\sum_i r_i$ be the aggregated transformed direction. Suppose each task loss has a β-Lipschitz continuous gradient. Then

$$f_i(\theta - \eta v) = f_i(\theta) - \eta\langle g_i, v\rangle + R_i, \qquad |R_i| \leq \frac{\beta}{2}\eta^2\|v\|^2. \tag{15}$$

Indeed, integration along the update segment gives

$$R_i = -\int_0^\eta \langle \nabla f_i(\theta - sv) - g_i, v\rangle\, ds,$$

and the remainder bound follows from gradient Lipschitz continuity.

The first-order change depends on alignment with the original task gradient. By the Cauchy–Schwarz inequality,

$$\langle g_i, v\rangle \geq \|g_i\|\left(\langle r_i, v\rangle - \|r_i - \bar{g}_i\|\,\|v\|\right).$$

Thus $-v$ is a strict common descent direction whenever

$$\langle r_i, v\rangle > \|r_i - \bar{g}_i\|\,\|v\| \qquad \text{for every } i.$$

Under this condition, all task losses decrease for

$$0 < \eta < \min_i \frac{2\langle g_i, v\rangle}{\beta\|v\|^2}.$$

This bound relates task-wise descent to the balance between alignment with the shared update and deviation from the original task direction.

G Multi-Task Equilibrium Relationship: A Cost–Benefit Analysis

G.1 Derivation of the Cost-Benefit Model

We examine the effect of rotation at a fixed iterate θ, with the task gradients and the EMA reference held fixed. The vector $\boldsymbol{\alpha}$ contains the task-specific rotation parameters. For the scalar analysis, α indexes an admissible path through this angular domain.

Assumption G.1 (Admissible Rotation Path)**.** The map $\alpha \mapsto \boldsymbol{\alpha}(\alpha)$ is continuously differentiable on $[0, \pi/2]$, satisfies $\boldsymbol{\alpha}(0) = \mathbf{0}$, and has nondecreasing components $\alpha_i(\alpha) \in [0, \pi/2]$.

Write $r_j(\alpha)$ for $r_j(\alpha_j(\alpha))$ and define

$$v(\alpha) = \frac{1}{T}\sum_{j=1}^{T} r_j(\alpha).$$

For fixed θ and η, define the first-order mean-loss model used in Eq. 1 by

$$\bar{f}(\alpha) := \frac{1}{T}\sum_{i=1}^{T} f_i(\theta) - \frac{\eta}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}\langle g_i, r_j(\alpha)\rangle.$$

Averaging Eq. 15 gives

$$\frac{1}{T}\sum_{i=1}^{T} f_i(\theta - \eta v(\alpha)) = \bar{f}(\alpha) + R(\alpha), \qquad |R(\alpha)| \le \frac{\beta}{2}\eta^2\|v(\alpha)\|^2.$$

At zero rotation, $r_j(0) = \bar{g}_j$ and $v(0) = \frac{1}{T}\sum_j \bar{g}_j$. Relative to this normalized-gradient baseline, define

$$\mathrm{CE}(\alpha) = \frac{\eta}{T^2}\sum_{i=1}^{T}\sum_{j\neq i}\langle g_i, r_j(\alpha) - \bar{g}_j\rangle,$$

$$\mathrm{FE}(\alpha) = \frac{\eta}{T^2}\sum_{i=1}^{T}\langle g_i, \bar{g}_i - r_i(\alpha)\rangle.$$

Subtracting the zero-rotation first-order model gives

$$\bar{f}(\alpha) = \bar{f}(0) - \mathrm{CE}(\alpha) + \mathrm{FE}(\alpha).$$

The actual mean-loss change relative to the zero-rotation update is

$$\frac{1}{T}\sum_i \big[f_i(\theta - \eta v(\alpha)) - f_i(\theta - \eta v(0))\big] = -\mathrm{CE}(\alpha) + \mathrm{FE}(\alpha) + R(\alpha) - R(0).$$

Since $\|v(\alpha)\| \le 1$, the residual difference satisfies $|R(\alpha) - R(0)| \le \beta\eta^2$.

Here $\mathrm{CE}(\alpha)$ measures the signed change in cross-task first-order contributions; positive values correspond to improved cross-task contributions. By Eq. 13,

$$\mathrm{FE}(\alpha) = \frac{\eta}{T^2}\sum_{i=1}^{T}\|g_i\|\big(1 - \cos\alpha_i(\alpha)\big) \ge 0. \tag{16}$$

The fidelity effect therefore quantifies the reduction in each task's own first-order contribution relative to its original normalized direction. Along the admissible path, it is nondecreasing.

G.2 Cost-Benefit Model Analysis

Following Section 2.2, to characterize a regime with diminishing alignment gains and increasing deviation costs, we model the benefit and cost along the scalar rotation path as

$$B(\alpha) = b\log(1+\alpha), \qquad C(\alpha) = c\big(\exp(k\alpha) - 1\big), \qquad b, c, k > 0.$$

Cost-Benefit Model. The corresponding modeled mean-loss profile is $\bar{f}(0) - B(\alpha) + C(\alpha)$ on $[0, \pi/2]$. The marginal benefit $B'(\alpha)$ decreases with rotation, whereas the marginal cost $C'(\alpha)$ increases.

Lemma G.2 (Convex–Concave Trade-off)**.** *B is strictly concave and C is strictly convex on $[0, \pi/2]$. Consequently, the modeled mean-loss profile is strictly convex.*

Proof. Direct differentiation gives

$$B'(\alpha) = \frac{b}{1+\alpha}, \qquad B''(\alpha) = -\frac{b}{(1+\alpha)^2} < 0,$$

and

$$C'(\alpha) = ck\exp(k\alpha), \qquad C''(\alpha) = ck^2\exp(k\alpha) > 0.$$

Thus $(-B + C)''(\alpha) > 0$ throughout the interval. □

Theorem G.3 (Unique Optimum and Existence of an Interior MER)**.** *The modeled mean-loss profile has a unique minimizer $\alpha^\star \in [0, \pi/2]$. It lies in $(0, \pi/2)$ if and only if*

$$b > ck, \qquad \frac{b}{1+\pi/2} < ck\exp(k\pi/2). \tag{17}$$

If $b \le ck$, then $\alpha^\star = 0$. If $b/(1+\pi/2) \ge ck\exp(k\pi/2)$, then $\alpha^\star = \pi/2$.

Proof. Continuity on the compact interval gives existence, and strict convexity gives uniqueness. The derivative $-b/(1+\alpha) + ck\exp(k\alpha)$ is strictly increasing. It has a zero in the interior exactly when its value at zero is negative and its value at $\pi/2$ is positive, which gives Eq. 17. The remaining sign cases give the stated endpoint minimizers. □

Theorem G.4 (MER First-Order Condition)**.** *Under Eq. 17, the unique interior minimizer satisfies*

$$B'(\alpha^\star) = C'(\alpha^\star), \qquad \frac{b}{1+\alpha^\star} = ck\exp(k\alpha^\star).$$

Proof. At an interior minimizer, differentiability implies $-B'(\alpha^\star) + C'(\alpha^\star) = 0$. Strict convexity makes this condition sufficient and identifies the unique minimizer. □

Corollary G.5 (Under- and Over-Correction Regimes)**.** *Suppose Eq. 17 holds. For $\alpha < \alpha^\star$, $B'(\alpha) > C'(\alpha)$ and increasing rotation decreases the modeled mean loss (under-correction). At $\alpha = \alpha^\star$, the marginal benefit and cost coincide. For $\alpha > \alpha^\star$, $B'(\alpha) < C'(\alpha)$ and increasing rotation increases the modeled mean loss (over-correction).*

The model characterizes MER as a balance of marginal benefit and cost. Its interior optimum captures a regime in which further alignment becomes less valuable than the additional deviation from task-specific directions.

G.3 Design of RGB Objective

The scalar model in Section 2.3 motivates controlling the amount of gradient modification. RGB implements an alignment–fidelity trade-off through the rotation objective

$$\min_{\boldsymbol{\alpha} \in [0,\pi/2]^T} \mathcal{L}(\boldsymbol{\alpha}),$$

$$\mathcal{L}(\boldsymbol{\alpha}) = \underbrace{\frac{1}{T(T-1)} \sum_{i<j} \big(1 - \langle r_i, r_j \rangle\big)}_{\text{Conflict term}} + \underbrace{\frac{1}{4T} \sum_i \|r_i - \bar{g}_i\|^2}_{\text{Fidelity term}}, \tag{18}$$

where $r_i = r_i(\alpha_i)$ and $T \ge 2$.

Fidelity term. Orthogonality gives the exact expression

$$\|r_i - \bar{g}_i\|^2 = (1 - \cos\alpha_i)^2 + \|w_i\|^2 \sin^2\alpha_i.$$

Under exact unit normalization, this becomes $2(1 - \cos\alpha_i)$, so the fidelity term is

$$\frac{1}{4T}\sum_i \|r_i - \bar{g}_i\|^2 = \frac{1}{2T}\sum_i (1 - \cos\alpha_i).$$

In this unit-rotation setting, the first-order fidelity effect in Eq. 16 has the same angular contributions, weighted by the original gradient magnitudes. In particular,

$$\frac{2\eta}{T}\min_i \|g_i\| \left(\frac{1}{4T}\sum_i \|r_i - \bar{g}_i\|^2\right) \le \mathrm{FE}(\alpha) \le \frac{2\eta}{T}\max_i \|g_i\| \left(\frac{1}{4T}\sum_i \|r_i - \bar{g}_i\|^2\right),$$

evaluated at $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\alpha)$. For unit original gradient magnitudes, the proportionality factor is exactly $2\eta/T$.

Conflict term. Minimizing the conflict term maximizes the sum of pairwise inner products among the transformed gradients. For unit r_i,

$$\frac{1}{T(T-1)}\sum_{i<j}(1 - \langle r_i, r_j\rangle) = \frac{T}{2(T-1)}(1 - \|v\|^2), \qquad v = \frac{1}{T}\sum_i r_i.$$

This expresses global directional agreement through the norm of the mean transformed gradient. The fidelity term relates this agreement to projections onto the original normalized gradients.

Proposition G.6 (Fidelity-Controlled Alignment Approximation)**.** *For* $\|\bar{g}_i\| = 1$ *and* $\|r_i\| \le 1$,

$$\left|\frac{1}{T(T-1)}\sum_{i\neq j}\langle \bar{g}_i, r_j\rangle - \frac{2}{T(T-1)}\sum_{i<j}\langle r_i, r_j\rangle\right| \le \sqrt{\frac{1}{T}\sum_i \|r_i - \bar{g}_i\|^2}.$$

Proof. The difference inside the absolute value equals

$$\frac{1}{T(T-1)}\sum_i \left\langle \bar{g}_i - r_i, \sum_{j\neq i} r_j \right\rangle.$$

Since $\|\sum_{j\neq i} r_j\| \le T - 1$, its absolute value is at most

$$\frac{1}{T}\sum_i \|r_i - \bar{g}_i\| \le \sqrt{\frac{1}{T}\sum_i \|r_i - \bar{g}_i\|^2},$$

where the last inequality follows from Cauchy–Schwarz. □

The bound is twice the square root of the fidelity term in Eq. 18. Thus the conflict term promotes agreement among the transformed gradients, while the fidelity term controls the discrepancy between this agreement and their cross-task projections onto the original normalized gradients. Together, the two terms provide a geometric implementation of the alignment–fidelity trade-off motivating MER.

H Derivation of Kernel-Based Objective

We derive the kernel formulation of the rotation objective in Eq. 5 and establish its equivalence to direct evaluation using the task-gradient vectors. Throughout the inner optimization, the basis vectors $\bar{g}_i$ and w_i are fixed.

Consider the RGB transformation

$$r_i(\alpha_i) = \cos\alpha_i\, \bar{g}_i + \sin\alpha_i\, w_i, \qquad \bar{g}_i, w_i, r_i(\alpha_i) \in \mathbb{R}^D, \qquad \alpha_i \in [0, \pi/2].$$

The task-specific angles form the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T) \in [0, \pi/2]^T$. For brevity, write $r_i = r_i(\alpha_i)$.

Define the stacked matrices

$$\bar{G} = \begin{bmatrix} \bar{g}_1^\top \\ \vdots \\ \bar{g}_T^\top \end{bmatrix}, \qquad W = \begin{bmatrix} w_1^\top \\ \vdots \\ w_T^\top \end{bmatrix}, \qquad R(\boldsymbol{\alpha}) = \begin{bmatrix} r_1(\alpha_1)^\top \\ \vdots \\ r_T(\alpha_T)^\top \end{bmatrix},$$

where $\bar{G}, W, R(\boldsymbol{\alpha}) \in \mathbb{R}^{T\times D}$. The Gram and cross-Gram matrices are

$$A = \bar{G}\bar{G}^\top, \qquad B = \bar{G}W^\top, \qquad C = W\bar{G}^\top, \qquad E = WW^\top,$$

with $A, B, C, E \in \mathbb{R}^{T\times T}$. Their entries satisfy

$$A_{ij} = \bar{g}_i^\top \bar{g}_j, \qquad B_{ij} = \bar{g}_i^\top w_j, \qquad C_{ij} = w_i^\top \bar{g}_j, \qquad E_{ij} = w_i^\top w_j.$$

Conflict term. The conflict term is

$$\frac{1}{T(T-1)} \sum_{i<j} (1 - r_i^\top r_j), \qquad T \geq 2.$$

The similarity between any two rotated gradients r_i and r_j is given by the inner product $r_i^\top r_j$. Expanding this term yields:

$$\begin{aligned} r_i^\top r_j &= (\cos\alpha_i \bar{g}_i + \sin\alpha_i w_i)^\top (\cos\alpha_j \bar{g}_j + \sin\alpha_j w_j) \\ &= (\cos\alpha_i \bar{g}_i^\top)(\cos\alpha_j \bar{g}_j) + (\cos\alpha_i \bar{g}_i^\top)(\sin\alpha_j w_j) \\ &\quad + (\sin\alpha_i w_i^\top)(\cos\alpha_j \bar{g}_j) + (\sin\alpha_i w_i^\top)(\sin\alpha_j w_j). \end{aligned}$$

Since $\cos\alpha_i$ and $\sin\alpha_j$ are scalars, they can be factored out of the vector inner products:

$$\begin{aligned} r_i^\top r_j &= (\cos\alpha_i \cos\alpha_j)(\bar{g}_i^\top \bar{g}_j) + (\cos\alpha_i \sin\alpha_j)(\bar{g}_i^\top w_j) \\ &\quad + (\sin\alpha_i \cos\alpha_j)(w_i^\top \bar{g}_j) + (\sin\alpha_i \sin\alpha_j)(w_i^\top w_j) \\ &= \cos\alpha_i \cos\alpha_j\, A_{ij} + \cos\alpha_i \sin\alpha_j\, B_{ij} \\ &\quad + \sin\alpha_i \cos\alpha_j\, C_{ij} + \sin\alpha_i \sin\alpha_j\, E_{ij}. \end{aligned}$$

Let $\mathbf{c} = \cos\boldsymbol{\alpha}$ and $\mathbf{s} = \sin\boldsymbol{\alpha}$. Then the full similarity matrix can be written compactly as:

$$\mathbf{S} = R(\boldsymbol{\alpha})R(\boldsymbol{\alpha})^\top = (\mathbf{c}\mathbf{c}^\top) \odot A + (\mathbf{c}\mathbf{s}^\top) \odot B + (\mathbf{s}\mathbf{c}^\top) \odot C + (\mathbf{s}\mathbf{s}^\top) \odot E, \tag{19}$$

where $\odot$ denotes element-wise multiplication. Since $S_{ij} = r_i^\top r_j$, the conflict term becomes

$$\frac{1}{T(T-1)} \sum_{i<j} (1 - S_{ij}).$$

Fidelity term. The fidelity term measures the squared Euclidean distance between each transformed vector and its original normalized task gradient:

$$\frac{1}{4T} \sum_{i=1}^{T} \|r_i - \bar{g}_i\|^2.$$

Expanding the squared norm yields

$$\|r_i - \bar{g}_i\|^2 = \|r_i\|^2 + \|\bar{g}_i\|^2 - 2r_i^\top \bar{g}_i.$$

The required quantities are available from the kernels:

$$\|r_i\|^2 = S_{ii}, \qquad \|\bar{g}_i\|^2 = A_{ii}, \qquad r_i^\top \bar{g}_i = c_i A_{ii} + s_i C_{ii}.$$

Therefore,

$$\|r_i - \bar{g}_i\|^2 = S_{ii} + A_{ii} - 2(c_i A_{ii} + s_i C_{ii}). \tag{20}$$

For an orthonormal rotation basis, $\|\bar{g}_i\| = \|w_i\| = 1$ and $w_i^\top \bar{g}_i = 0$. In this setting, $\|r_i\| = 1$ and $r_i^\top \bar{g}_i = \cos\alpha_i$, so it simplifies to

$$\|r_i - \bar{g}_i\|^2 = 2(1 - \cos\alpha_i).$$

Consequently, the fidelity term takes the angular form

$$\frac{1}{4T}\sum_{i=1}^{T} \|r_i - \bar{g}_i\|^2 = \frac{1}{2T}\sum_{i=1}^{T}(1 - \cos\alpha_i).$$

Complete kernel objective. Combining the two terms gives

$$\begin{aligned}\mathcal{L}_{\text{kernel}}(\boldsymbol{\alpha}) = \frac{1}{T(T-1)}\sum_{i<j}(1 - S_{ij}) \\ + \frac{1}{4T}\sum_{i=1}^{T}\left[S_{ii} + A_{ii} - 2(c_i A_{ii} + s_i C_{ii})\right].\end{aligned} \tag{21}$$

Proposition H.1 (Equivalence of the Kernel Objective)**.** *For fixed basis vectors $\bar{g}_i, w_i$, the kernel objective in Eq. 21 equals the original squared-distance RGB objective for every feasible $\boldsymbol{\alpha}$:*

$$\mathcal{L}_{\text{kernel}}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}).$$

Their gradients with respect to the rotation parameters also coincide:

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}_{\text{kernel}}(\boldsymbol{\alpha}) = \nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}).$$

Proof. Eq. 19 expresses every pairwise inner product $r_i^\top r_j$ exactly in terms of A, B, C, E and the rotation parameters. Eq. 20 provides an exact expression for each squared fidelity distance using the same matrices. Substituting these identities into Eq. 5 yields Eq. 21, establishing equality of the two objective functions.

Both expressions are differentiable functions of $\boldsymbol{\alpha}$ with fixed basis vectors. Their pointwise equality therefore implies equality of their gradients with respect to $\boldsymbol{\alpha}$. □

The kernel formulation evaluates the inner products required by the objective through precomputed Gram matrices. Each transformed vector remains in $\text{span}\{\bar{g}_i, w_i\}$, a subspace of dimension at most two. No spectral truncation, low-rank approximation, or feature compression is introduced. After constructing the Gram matrices, objective evaluation and differentiation operate on task-space quantities, avoiding repeated operations on the D-dimensional task vectors.

I Angular Sweeps Across Datasets and Training Checkpoints

We investigate how the amount of gradient rotation affects short-horizon task optimization across Cityscapes, NYUv2, and QM9. For each dataset, we perform angular sweeps at training iterations 100, 2000, and 5000, yielding nine experiments. These sweeps examine the dependence of the preferred rotation amount on both the dataset and the training stage.

Experimental protocol. We use the FAMO codebase for two-task Cityscapes, LibMTL for NYUv2, and Nash-MTL for QM9, following the corresponding benchmark setups. For each dataset, we save the model at training iterations 100, 2000, and 5000 and sweep the rotation parameter from 0° to 90° in 1° increments. At each candidate angle, we apply the prescribed rotation to the task gradients and perform a five-step lookahead simulation. Each simulation starts from the saved checkpoint, without permanently modifying the original model weights. We record the geometric conflict and fidelity quantities at the initial lookahead step and the empirical mean task loss after the five updates. For a fixed dataset and checkpoint, let $\theta^{(5)}(\alpha)$ denote the parameters obtained after the five-step lookahead simulation, and define

$$\ell_5(\alpha) := \frac{1}{T}\sum_{i=1}^{T} f_i\big(\theta^{(5)}(\alpha)\big).$$

The empirical mean-loss change is

$$\Delta\ell_5(\alpha) := \ell_5(\alpha) - \ell_5(0).$$

The quantity labeled $\Delta\mathcal{L}$ in the figures corresponds to $\Delta\ell_5(\alpha)$. The baseline is the five-step simulation with zero rotation; thus, $\Delta\ell_5(\alpha) < 0$ indicates a lower mean loss than the no-rotation update. Angles are displayed in degrees in the figures and expressed in radians in the analytical formulas. Each sweep uses the common-angle parameterization

$$\boldsymbol{\alpha}(\alpha) = \alpha\mathbf{1}, \qquad \alpha \in [0, \pi/2],$$

where $\mathbf{1} \in \mathbb{R}^T$ is the all-ones vector.

Geometric measurements. For the task vectors $r_i(\alpha)$ at the initial lookahead step, define

$$\mathcal{C}_{\text{geo}}(\alpha) := \frac{1}{T(T-1)}\sum_{i<j}\big[1 - r_i(\alpha)^\top r_j(\alpha)\big].$$

The measured conflict-reduction benefit and fidelity distortion are

$$B_{\text{geo}}(\alpha) := \mathcal{C}_{\text{geo}}(0) - \mathcal{C}_{\text{geo}}(\alpha),$$

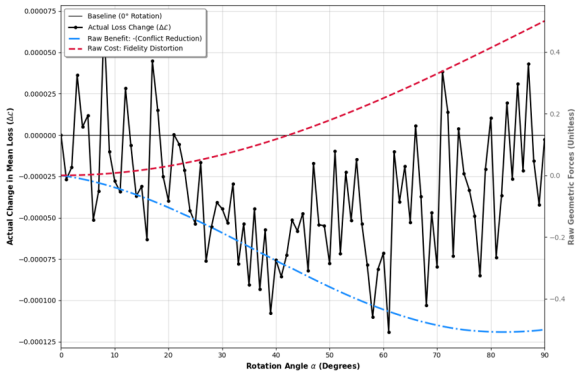
$$C_{\text{geo}}(\alpha) := \frac{1}{4T}\sum_{i=1}^{T}\|r_i(\alpha) - \bar{g}_i\|^2.$$

The figures plot $-B_{\text{geo}}(\alpha)$ in blue and $C_{\text{geo}}(\alpha)$ in red on the secondary axis. A downward movement of the blue curve therefore indicates greater conflict reduction. The black curve reports the five-step empirical loss change on the primary axis. The geometric quantities characterize the initial transformation, while the empirical loss reflects the subsequent five-step optimization trajectory.

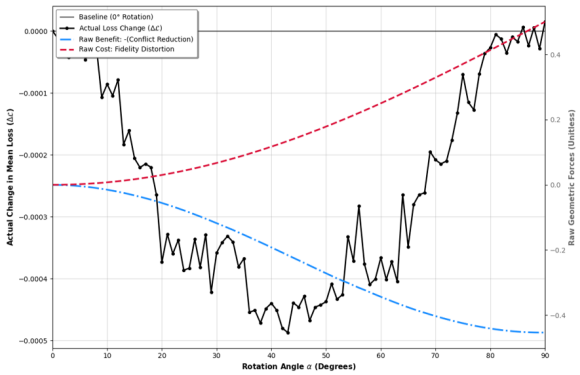
Cityscapes. Figure 3 presents the two-task Cityscapes experiment with semantic segmentation and depth estimation. At iteration 100, the empirical loss curve exhibits substantial fluctuations, making the location of a stable preferred angle difficult to identify. At iterations 2000 and 5000, the loss follows a broad U-shaped profile: moderate rotation improves upon the zero-rotation baseline, the lowest observed losses occur near 40°–43°, and larger rotations increase the loss toward the baseline or above it. The similar locations of these minima across the later checkpoints indicate a persistent preference for an intermediate rotation amount in this setting.

NYUv2. Figure 4 shows a preference for small rotations, with the lowest observed losses occurring within approximately the first few degrees. At iterations 2000 and 5000, the loss initially falls below the zero-rotation baseline and then generally increases as rotation becomes larger. Consequently, the small-angle minimum is followed by an extended region of degraded performance, rather than a broad minimum near the middle of the angular interval. The iteration-100 curve also favors a small angle, although its subsequent trajectory is more irregular and nonmonotonic.

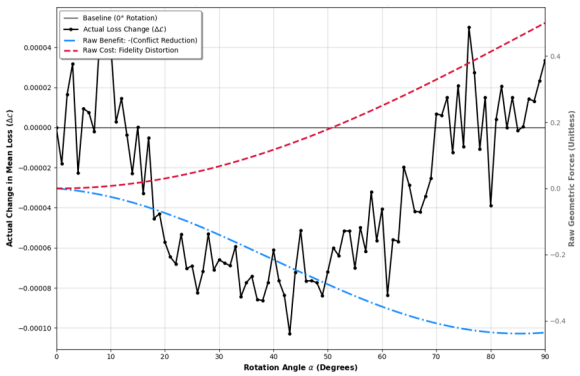
QM9. Figure 5 exhibits a pronounced dependence on training stage. At iteration 100, the loss curve is highly variable and contains substantial improvements at intermediate and larger angles. At iterations 2000 and 5000, the preferred rotation shifts to the near-zero region. Larger rotations generally increase the mean loss, although the iteration-2000 curve retains substantial local fluctuations. The later checkpoints therefore favor conservative gradient modification in this setting.



(a) Iteration 100.

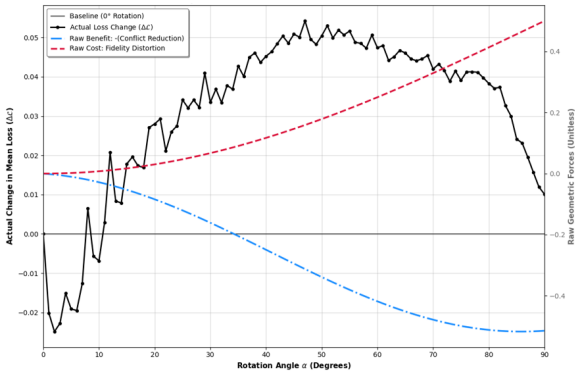


(b) Iteration 2000.

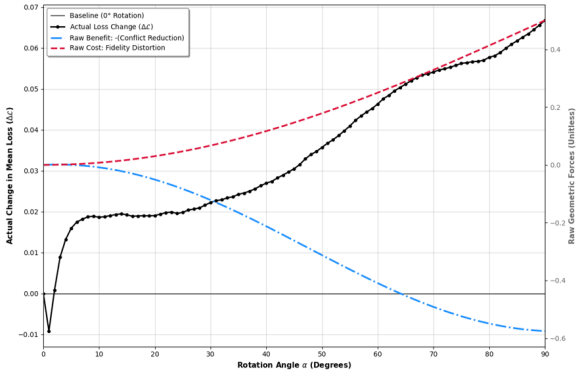


(c) Iteration 5000.

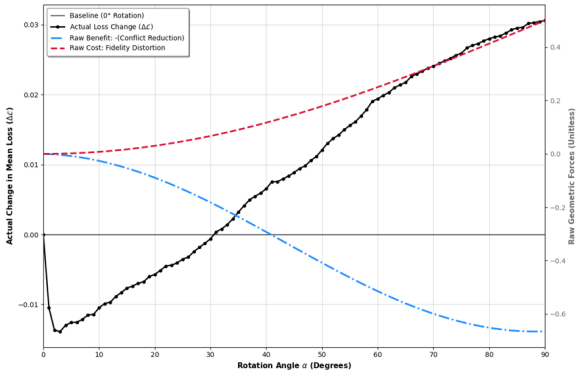
Figure 3: Angular sweeps on Cityscapes. The black curve shows the five-step mean-loss change relative to zero rotation on the left axis. The blue dash-dot and red dashed curves show negative conflict-reduction benefit and fidelity distortion, respectively, on the right axis. Later checkpoints exhibit a broad U-shaped loss profile, with minima near $40°$–$43°$.



(a) Iteration 100.

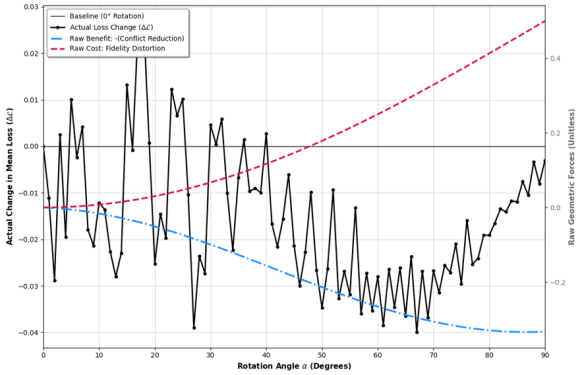


(b) Iteration 2000.

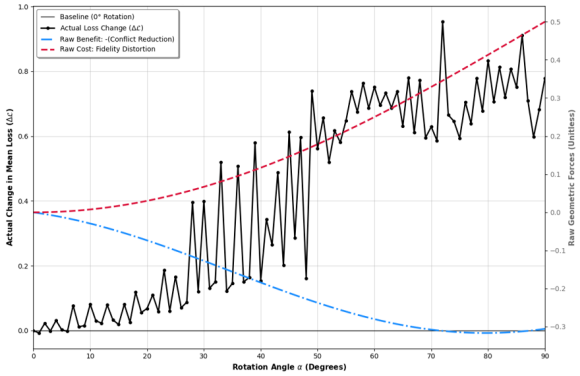


(c) Iteration 5000.

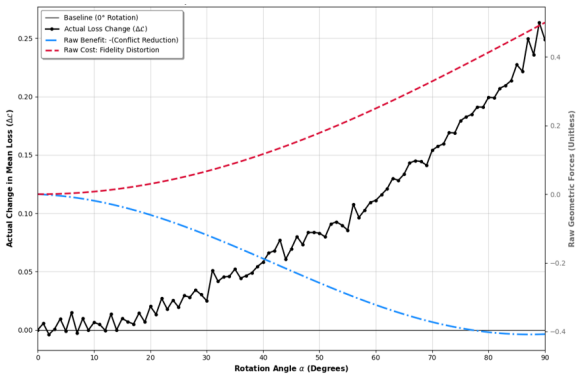
Figure 4: Angular sweeps on NYUv2. Axes and curve definitions follow Figure 3. The lowest observed losses occur at small rotation angles. At the later checkpoints, increasing rotation beyond this region generally raises the empirical mean loss.



(a) Iteration 100.



(b) Iteration 2000.



(c) Iteration 5000.

Figure 5: Angular sweeps on QM9. Axes and curve definitions follow Figure 3. The early checkpoint exhibits substantial fluctuations. At the later checkpoints, the lowest observed losses occur near zero rotation, while larger rotations generally increase the empirical mean loss.

Relationship to the cost–benefit model. Across the sweeps, the geometric conflict-reduction curves generally flatten at larger angles, while fidelity distortion increases with rotation. Some conflict-reduction curves also show a slight reversal near the upper end of the interval. For the unit-rotation setting with a common angle, the fidelity quantity has the explicit form

$$C_{\text{geo}}(\alpha) = \frac{1}{2}(1 - \cos\alpha), \qquad C''_{\text{geo}}(\alpha) = \frac{1}{2}\cos\alpha \geq 0, \quad \alpha \in [0, \pi/2].$$

This identity explains the convex geometric distortion profile over the prescribed angular range.

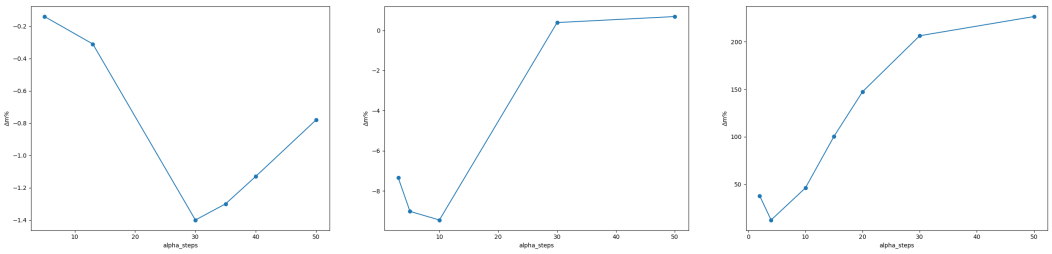
The analytical functions $B(\alpha) = b\log(1+\alpha)$ and $C(\alpha) = c(\exp(k\alpha) - 1)$ in Section 2.2 model diminishing marginal benefit and increasing marginal cost. They provide tractable instances of the concave-benefit and convex-cost structure. The angular sweeps examine the resulting trade-off empirically; these analytical functions are not fitted to the plotted geometric measurements. Within the scalar model, strict convexity of $C - B$ gives a unique minimizer, and the boundary derivative conditions in Theorem G.3 determine whether it lies in the interior.

Implications for gradient modification. The later checkpoints consistently favor intermediate rotation on Cityscapes and much smaller rotation on NYUv2 and QM9. In all nine displayed sweeps, the 90° endpoint has a higher empirical loss than the lowest observed value. These results support controlling the amount of gradient modification and show that the preferred rotation depends on the dataset and training stage.

The empirically determined inner-loop budgets, $\alpha_{\text{steps}} = 30$ for Cityscapes, 10 for NYUv2, and 4 for QM9, likewise differ across datasets. These values specify the number of inner optimization iterations; the angular sweeps directly probe the response to prescribed rotation amounts. Together, the experiments motivate dataset-specific control of gradient modification and provide empirical support for the alignment–fidelity trade-off underlying MER.

J Ablation Study: Alpha Steps Sensitivity

We investigate the sensitivity of RGB to α_{steps}, the number of inner optimization iterations used to update the task-specific rotation parameters $\boldsymbol{\alpha}$. This hyperparameter determines the optimization budget allocated to the rotation objective and influences the angles returned by the inner loop. We evaluate different budgets on Cityscapes, NYUv2, and QM9, using the same random seed across budget settings within each dataset.

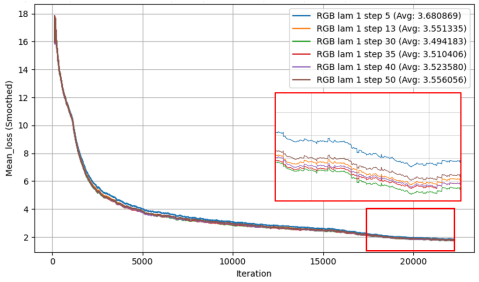


(a) Cityscapes (b) NYUv2 (c) QM9

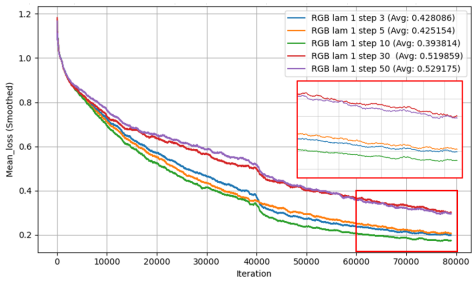
Figure 6: Comparison of performance for different inner-iteration budgets across three datasets.

Multi-task performance ($\Delta m\%$). Figure 6 shows a U-shaped performance trend across the tested budgets: the relative performance drop $\Delta m\%$ initially decreases and subsequently increases as α_{steps} grows. Among the evaluated settings, the lowest $\Delta m\%$ is obtained with $\alpha_{\text{steps}} = 30$ for Cityscapes, 10 for NYUv2, and 4 for QM9. These results indicate that the preferred inner-iteration budget depends on the task configuration.

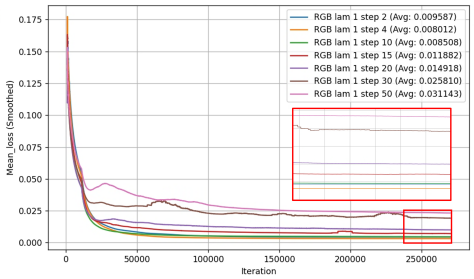
The observed sensitivity is consistent with the alignment–fidelity trade-off motivating MER. A limited optimization budget can restrict the adjustment of task gradients, whereas additional iterations allow the inner optimizer to explore larger modifications. In the evaluated settings, increasing the budget beyond the best-performing configuration degrades multi-task performance. This behavior supports controlling the extent of gradient modification when selecting the inner-loop budget.



(a) Cityscapes

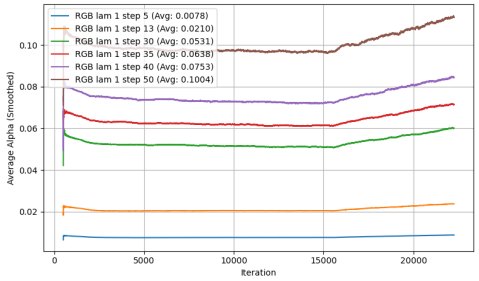


(b) NYUv2

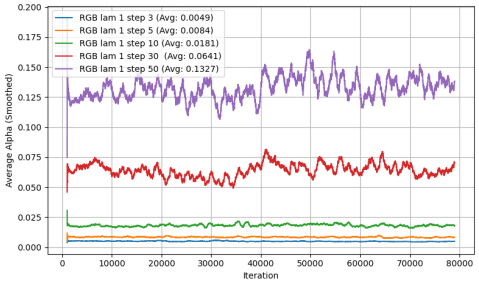


(c) QM9

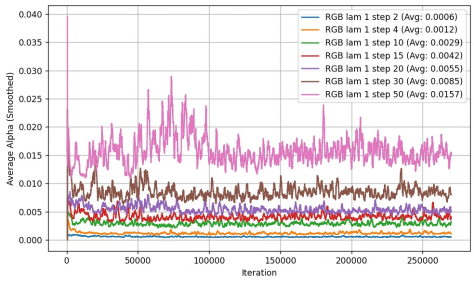
Figure 7: Comparison of mean loss for different inner-iteration budgets across three datasets.



(a) Cityscapes



(b) NYUv2



(c) QM9

Figure 8: Comparison of average alpha for different inner-iteration budgets across three datasets.

Training dynamics. Figure 7 compares the mean training loss across inner-iteration budgets. The budgets yielding lower $\Delta m\%$ also tend to produce lower and more stable training-loss trajectories. Larger budgets beyond the preferred range can instead lead to higher loss plateaus or greater fluctuations. These observations complement the performance comparison and indicate that the inner-loop budget affects both training dynamics and the final multi-task results.

Evolution of the rotation angles. To examine the adjustments produced by different budgets, we track the returned task-specific angles $\tilde{\boldsymbol{\alpha}}_t = (\tilde{\alpha}_{1,t}, \ldots, \tilde{\alpha}_{T,t})$ at outer iteration t. Their average is

$$\bar{\alpha}_t := \frac{1}{T} \sum_{i=1}^{T} \tilde{\alpha}_{i,t}.$$

Figure 8 shows that larger α_{steps} values are generally associated with larger average rotation angles in these experiments. This provides an empirical connection between the inner-iteration budget and the extent of gradient modification. The returned angles depend jointly on the budget, the rotation objective, the inner optimizer, and the current task gradients.

Practical selection of α_{steps}. Our sensitivity analysis shows that the appropriate inner-optimization budget is dataset dependent, and we therefore do not claim a universal value of α_{steps}. The values reported in the main experiments are identified through the sensitivity analysis in Figure 6. For practical use on a new dataset, α_{steps} should be treated as a standard hyperparameter and selected on a held-out validation set e.g., begin with a small candidate from the set of $\{5, 10, 30, 50\}$. Then, we gradually increase the candidate to monitor the condition of the validation $\Delta m\%$ and stop when the validation $\Delta m\%$ begins to increase.

K Tolerance Limit: Fidelity Sensitivity Analysis

Aggregate fidelity tolerance. RGB balances agreement among task gradients with fidelity to their original directions. We investigate this trade-off by examining how empirical mean task loss changes with the measured fidelity distortion across different inner-iteration budgets. We use $\mathcal{F}_{\text{tol}}$ to denote an empirical aggregate fidelity threshold: a transition region where additional distortion becomes associated with increasing mean loss.

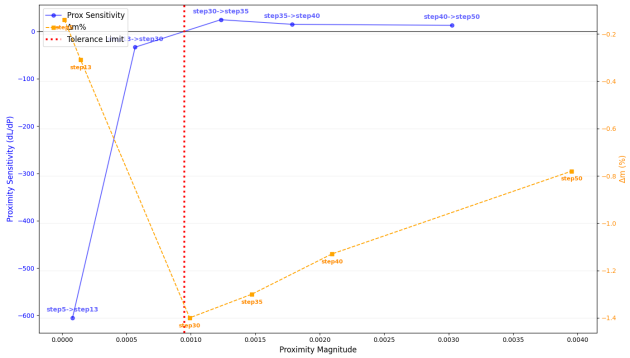
Experimental setting. We evaluate a sequence of inner-iteration budgets α_{steps} on Cityscapes, NYUv2, and QM9. Changing this budget affects the task-specific angles returned by the inner optimization and can therefore change both gradient alignment and fidelity distortion. The arrows in Figure 9 connect the evaluated configurations, such as $\alpha_{\text{steps}} = 5$ and $\alpha_{\text{steps}} = 10$.

Fidelity sensitivity. Index the evaluated inner-iteration budgets by q. Let ℓ_q denote the recorded empirical mean task loss and $\mathcal{F}_q$ the corresponding fidelity distortion, measured by the scaled mean squared Euclidean distance between the adjusted and original task-gradient directions. For adjacent configurations with $\mathcal{F}_q \neq \mathcal{F}_{q-1}$, define

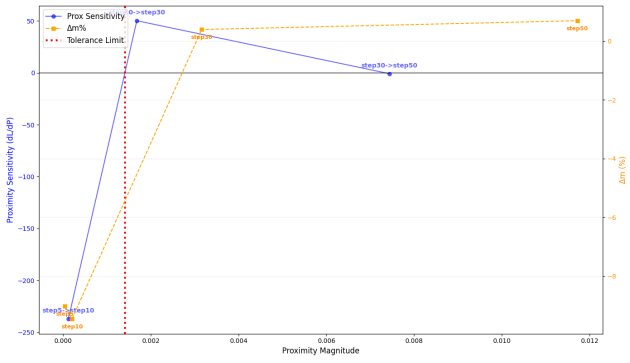
$$S_{\mathcal{F},q} := \frac{\ell_q - \ell_{q-1}}{\mathcal{F}_q - \mathcal{F}_{q-1}}. \tag{22}$$

This finite-difference quantity measures the change in empirical mean loss per unit change in fidelity distortion. For transitions satisfying $\mathcal{F}_q > \mathcal{F}_{q-1}$, negative sensitivity indicates that greater distortion is accompanied by lower mean loss, whereas positive sensitivity indicates that greater distortion is accompanied by higher mean loss. A transition from negative to positive sensitivity therefore identifies an empirical region in which the loss response to additional distortion changes sign.

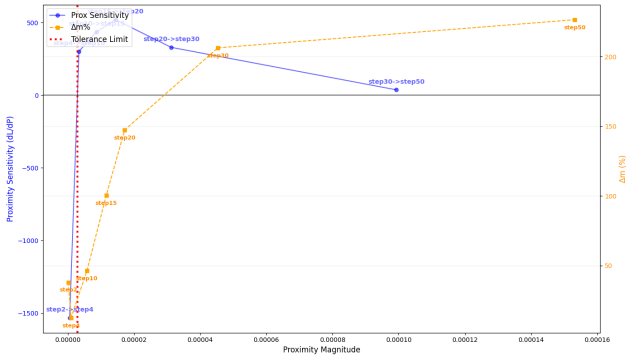
Observed transition. Figure 9 shows that the relationship between fidelity distortion and empirical mean loss varies across the evaluated configurations. The transition from negative to positive sensitivity provides an operational estimate of $\mathcal{F}_{\text{tol}}$ within the sampled range. For Cityscapes, this transition occurs near $\mathcal{F}_{\text{tol}} \approx 0.0010$, whereas QM9 exhibits a transition at a smaller fidelity magnitude. These observations indicate that the aggregate tolerance to gradient modification depends on the benchmark configuration.



(a) **Cityscapes**



(b) **NYUv2**



(c) **QM9**

Figure 9: Fidelity sensitivity across inner-iteration budgets. Arrows connect evaluated α_{steps} settings, and the horizontal axis reports fidelity distortion. For transitions with increasing distortion, negative $S_{\mathcal{F},q}$ indicates decreasing empirical mean loss, while positive $S_{\mathcal{F},q}$ indicates increasing mean loss. The red dotted line marks the approximate aggregate fidelity threshold $\mathcal{F}_{\text{tol}}$ inferred from the observed sensitivity transition.

Relationship to MER. The sensitivity transitions are consistent with the alignment–fidelity trade-off motivating MER. Within the evaluated range, additional gradient modification can accompany improved optimization, while further distortion can accompany increasing loss. This analysis complements the performance sensitivity results in Appendix J by characterizing the empirical mean-loss response in terms of measured fidelity distortion.

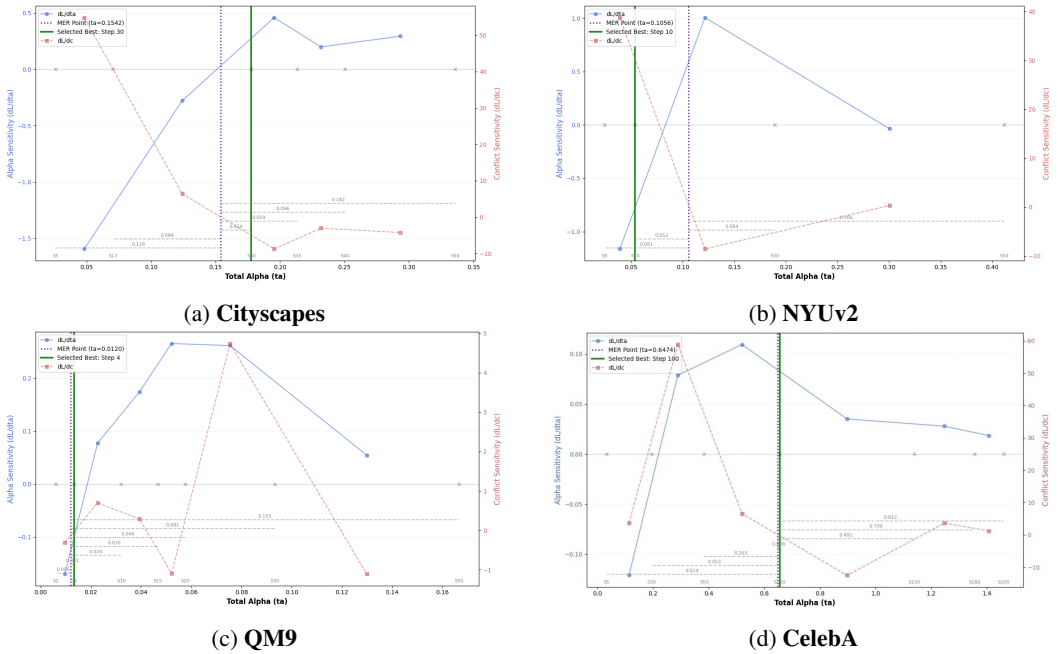
L MER DISTANCES AND PERFORMANCE ANALYSIS

In this section, we empirically validate MER and evaluate its multi-task performance across four different datasets. We perform a series of independent experiments across a range of fixed inner-iteration budgets (α_{steps}) for each dataset. For every experimental run, we record three primary metrics at the point of convergence:

- **Total Alpha ($t\alpha$):** The sum magnitude of gradient rotation across all tasks for a given inner-iteration budget (α_{steps}). Specifically, this metric corresponds to the mean Euclidean distance between the adjusted and original gradient directions across all tasks, serving as our empirical measure for the fidelity term.
- **Conflict Magnitude (c):** The empirical measurement of the conflict term as formulated in the optimization objective.
- **Mean Loss across all tasks (L):** The logged empirical average task loss.

Subsequently, the MER state followed Eq. 4 is defined as the point where the marginal benefit of conflict reduction exactly offsets the marginal cost of fidelity distortion ($dL/dc = dL/dt\alpha$). Due to the discrete nature of the conducted experiments, we apply linear interpolation between adjacent empirical observation points to derive the MER value alongside its corresponding adjustment magnitude, $t\alpha_{MER}$. We then define the metric **Distance to MER** ($\mathcal{D}_{MER}$) as follows:

$$\mathcal{D}_{MER} = |t\alpha_{current} - t\alpha_{MER}|$$



(a) **Cityscapes** (b) **NYUv2**

(c) **QM9** (d) **CelebA**

Figure 10: Relationship between total rotation, distance to the empirical MER reference, and multi-task performance on Cityscapes, NYUv2, QM9, and CelebA.

The overall analysis is illustrated in Figure 10, showing a consistent trend where the best overall task performance ($\Delta m\%$) generally yields the lowest $\mathcal{D}_{MER}$. Furthermore, we identify three distinct regimes:

- **The Under-Correction Regime:** Characterized by a high $\mathcal{D}_{MER}$. In this stage, the marginal benefit (dL/dc) is high, whereas the fidelity cost ($dL/dt\alpha$) is negligible. Sub-optimal performance occurs because significant gradient conflict remains unresolved.
- **The MER Point (Optimal Balance):** As α increases, dL/dc diminishes while $dL/dt\alpha$ accelerates. At the optimal step (e.g., Step 30 for Cityscapes), $\mathcal{D}_{MER}$ is minimized. The operating point closest to the estimated MER coincides with the lowest observed mean loss and the best observed $\Delta m\%$ among the tested settings.
- **The Over-Correction Regime:** Beyond equilibrium, the $\mathcal{D}_{MER}$ widens as $dL/dt\alpha$ spikes. The algorithm begins destroying task-specific information to enforce geometric alignment, leading to a precipitous drop in performance.

This analysis complements the fidelity sensitivity study in Appendix K by expressing the observed trade-off in terms of total rotation and its distance from an empirical reference.

M Miss-Correction of Other Gradient Manipulation Methods

To validate the theoretical miss-correction regimes in practice, we examine the relationship between gradient modification and task optimization by comparing geometric conflict, fidelity distortion, and empirical mean task loss on NYUv2 and QM9. This analysis complements the rotation ablations by assessing how different gradient manipulation methods balance directional agreement with deviation from the original task gradients.

Tables 3 and 4 report these quantities for GradDrop, GradVac, PCGrad, and RGB. GradVac achieves the lowest conflict value among the compared methods on both benchmarks: 0.342245 on NYUv2 and 0.225787 on QM9. However, its mean task loss exceeds that of RGB on both datasets. On QM9, GradVac records a fidelity distortion of 0.107321 and the highest reported mean loss, 0.083342.

PCGrad also attains lower conflict values than RGB, but exhibits greater fidelity distortion: 0.007641 on NYUv2 and 0.046394 on QM9. Its corresponding mean losses, 0.445967 and 0.034706, are higher than those of RGB. GradDrop exhibits the largest fidelity distortion among the compared methods on both datasets and likewise has higher mean losses than RGB.

RGB achieves the lowest reported mean loss and fidelity distortion on both benchmarks. On NYUv2, it obtains a mean loss of 0.388934 and fidelity distortion of 2.07×10^{-4}, while retaining the highest conflict value among the compared methods. On QM9, it obtains a mean loss of 0.015436 and fidelity distortion of 6.21×10^{-5}, with a conflict value higher than those of GradVac and PCGrad.

These comparisons show that lower geometric conflict alone does not consistently indicate better task optimization. The combination of low fidelity distortion and low task loss observed for RGB is consistent with the alignment–fidelity trade-off motivating MER.

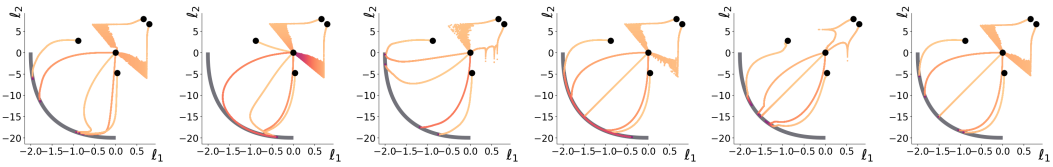
Table 3: Average losses, conflict, and fidelity for different gradient manipulation methods in NYU.

Method	Avg Loss ↓	Conflict ↓	Fidelity ↓
GradDrop	0.472945	0.431341	0.138928
GradVac	0.445305	**0.342245**	0.020390
PCGrad	0.445967	0.399961	0.007641
RGB	**0.388934**	0.441383	**2.07E-04**

Table 4: Average losses, conflict, and fidelity for different gradient manipulation methods in QM9.

Method	Avg Loss ↓	Conflict ↓	Fidelity ↓
GradDrop	0.035939	0.453621	0.110055
GradVac	0.083342	**0.225787**	0.107321
PCGrad	0.034706	0.322144	0.046394
RGB	**0.015436**	0.425304	**6.21E-05**

N THE CONVERGENCE ANALYSIS BASED ON TOY EXPERIMENT



(a) PCGrad (b) CAGrad (c) MGDA (d) Nash-MTL (e) FAMO (f) RGB (Ours)

Figure 11: Visualization of optimization trajectories on a two-task toy problem. Each subplot corresponds to a different multi-task learning (MTL) method: PCGrad, CAGrad, MGDA, Nash-MTL, FAMO ($\gamma = 0.0$), and our proposed RGB. The axes denote the per-task losses ℓ_1 and ℓ_2, and the Pareto front is illustrated in dark gray. All experiments were implemented using the publicly released FAMO repository and executed for 50,000 steps under identical settings.

As shown in Figure 11, PCGrad, CAGrad, and MGDA all successfully drive their trajectories onto the Pareto front. In contrast, Nash-MTL and FAMO achieve smooth convergence with trajectories neatly landing on the front. Our proposed RGB method also converges cleanly to the front. Importantly, RGB is a *gradient-based* method, yet it achieves stable and consistent convergence comparable to weighting-based methods. To quantify this behavior, we compare RGB and Nash-MTL using five fixed initializations per method. We report the final objective values, the final Euclidean distance to the numerical Pareto front, and the first iteration at which that distance falls below 5% of its initial value.

Table 5: Comparison of RGB and Nash-MTL metrics across different initializations.

Init	RGB (ℓ_1, ℓ_2)	RGB dist-to-front	RGB steps	Nash-MTL (ℓ_1, ℓ_2)	Nash dist-to-front	Nash steps
0	(-1.0766, -18.0595)	0.0030	10788	(-0.3812, -19.8261)	0.0025	9984
1	(-1.5096, -15.0876)	0.0115	4677	(-0.3793, -19.8281)	0.0018	4309
2	(-1.8044, -10.7995)	0.0018	10510	(-0.3790, -19.8284)	0.0015	9341
3	(-0.9197, -18.6876)	0.0081	4696	(-0.3792, -19.8281)	0.0017	4340
4	(-1.9583, -5.6333)	0.0080	1948	(-0.3880, -19.8285)	0.0013	2196
Mean	–	0.0065 ± 0.0040	6524	–	0.0018 ± 0.0005	6034

Empirical convergence behavior. Both methods reach the prescribed distance threshold from all five initializations. RGB has a mean first hitting iteration of 6524, compared with 6034 for Nash-MTL. Their mean final distances are 0.0065 and 0.0018, respectively. Thus, both methods approach the numerical front within the training budget, with Nash-MTL achieving a smaller mean final distance in this experiment.

The final objective values reveal different task trade-offs. Nash-MTL's endpoints cluster near $(\ell_1, \ell_2) = (-0.38, -19.83)$. RGB reaches more dispersed endpoints, with ℓ_1 ranging from approximately -1.96 to -0.92. These results show that proximity to the same Pareto front can accompany substantially different allocations of performance across the two tasks.

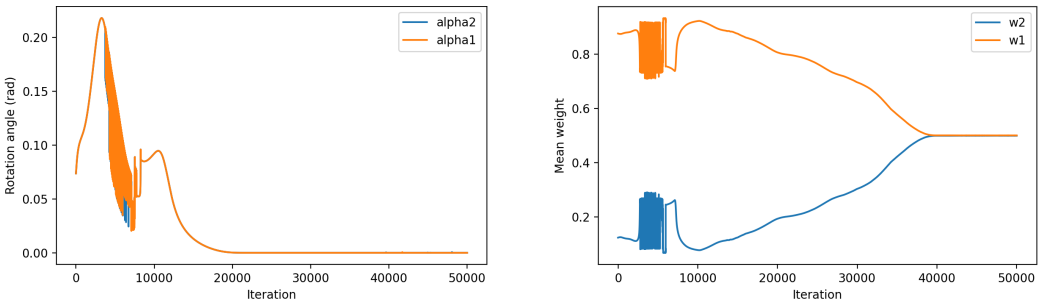
Rotation and reweighting dynamics. Figure 12 compares the internal dynamics of the two methods. RGB jointly adjusts task-specific rotation angles and uniformly aggregates the transformed task vectors. Nash-MTL constructs its update through task-dependent weights. Writing

$$G = [g_1, \ldots, g_T] \in \mathbb{R}^{D \times T},$$

its aggregated gradient vector has the form

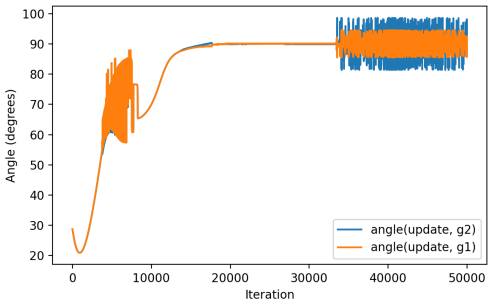
$$v_{\text{Nash}} = G\boldsymbol{\omega} = \sum_{i=1}^{T} \omega_i g_i,$$

where $\boldsymbol{\omega}$ denotes the task-weight vector.

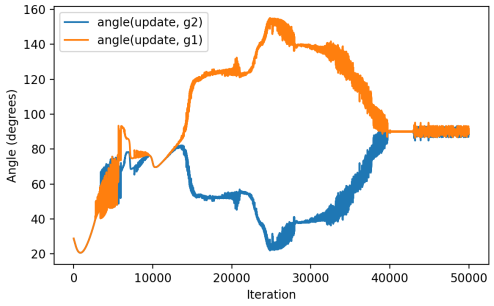


(a) RGB rotation angles.

(b) Nash-MTL weights.



(c) Angle(update, g_1/g_2) for RGB.



(d) Angle(update, g_1/g_2) for Nash-MTL.

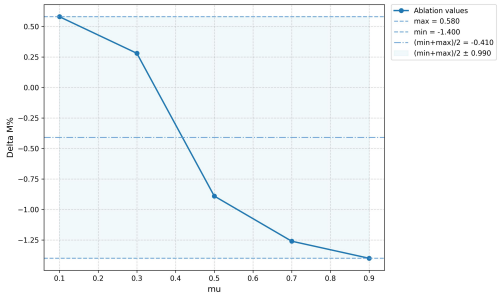
Figure 12: Optimization dynamics in the toy FAMO experiment: rotation (RGB), reweighting (Nash-MTL), and update–gradient alignment.

The displayed RGB rotation angles rise during the early iterations and subsequently decline toward zero. This trajectory indicates that the amount of gradient modification decreases as training progresses in the illustrated run. Nash-MTL instead adjusts its relative task contributions through the evolving weight vector.

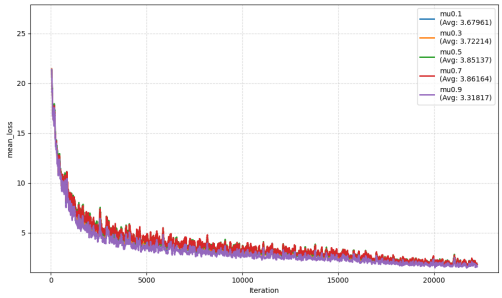
Update–gradient alignment. The angles between the RGB update and each task gradient converge to $\approx 90°$, so the shared update becomes nearly orthogonal to both individual gradients and thus aligned with a common descent direction close to a Pareto-stationary solution. Unlike Nash-MTL, RGB maintains a constant, equal weight across all tasks throughout training. Nash-MTL instead achieves compromise by reweighting: its task weights start strongly imbalanced, traverse a long transient phase, and finally converge nearly $0.5{:}0.5$ respectively, while the angles between the update and each task gradient oscillate widely and frequently cross $90°$, reflecting a bargaining-like process that alternately favors one task and then the other.

The alignment trajectories provide a geometric diagnostic of the evolving updates. Together with the distance-to-front measurements, they show that rotation and reweighting can produce similar overall trajectories toward the Pareto front, while exhibiting different internal dynamics and reaching different final task trade-offs.

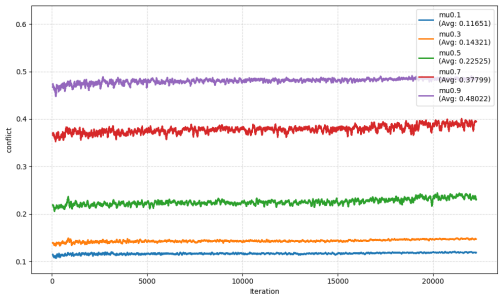
O ABLATION STUDY ON COEFFICIENTS μ



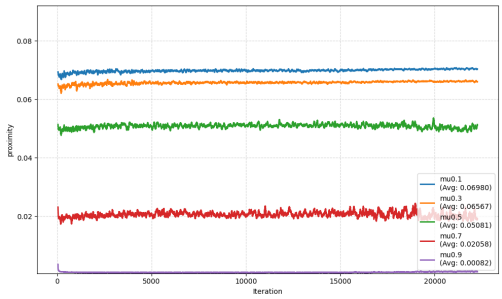
(a) Δm% for μ



(b) Loss Over Iteration



(c) Conflict Over Iteration



(d) Fidelity Over Iteration

Figure 13: Ablation study of coefficient μ on Cityscapes dataset.

We analyze the impact of the EMA momentum parameter by sweeping $\mu \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Figure 13a shows that $\Delta m\%$ decreases as μ increases across the tested settings, reaching its minimum at $\mu = 0.9$. This highlights the necessity of a slowly evolving global direction d_t; a larger momentum ensures that d_t tracks a long-term consensus of the task gradients rather than reacting to the instantaneous noise of individual iterations.

The conflict and fidelity trajectories in Figures 13c and 13d provide complementary information about the resulting gradient modifications. Lower momentum values, particularly $\mu = 0.1$ and $\mu = 0.3$, produce lower geometric conflict but greater fidelity distortion. These configurations therefore achieve stronger alignment while introducing larger deviations from the original task-gradient directions. Their higher $\Delta m\%$ indicates that the additional alignment does not translate into better multi-task performance in this experiment. In this regime, the rotation operator aggressively chasing short-term fluctuations because a small μ causes d_t to shift rapidly.

Conversely, $\mu = 0.9$ produces the lowest fidelity distortion among the evaluated settings while retaining a higher level of geometric conflict. Its lower $\Delta m\%$ suggests that preserving more of the original task directions is beneficial in this setting. By leveraging past information to smooth gradient noise, the model prevents d_t from shifting rapidly and captures a more accurate requirement for task alignment. These observations support the use of $\mu = 0.9$ in our experiments and suggest that a high EMA momentum can provide a favorable alignment–fidelity trade-off.

Relationship to MER. The results are consistent with the alignment–fidelity trade-off motivating MER. Changing μ alters the temporal evolution of the reference direction and consequently the gradient modifications produced by RGB. In the evaluated configurations, stronger geometric alignment accompanies greater distortion and poorer performance, whereas greater reliance on historical directions accompanies lower distortion and better performance. This ablation therefore highlights the role of the EMA reference in controlling gradient modification during training.

P EXPERIMENT DETAILS

Table 6: Full results on QM9. All metrics are MAE (lower is better). MR is the mean rank across the 11 metrics; smaller is better.

Method	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\langle R^2 \rangle$	ZPVE	U_0	U	H	G	C_v	MR ↓	$\Delta m\%$ ↓
	MAE ↓												
STL	0.067	0.181	60.57	53.91	0.502	4.53	58.80	64.20	63.80	66.20	0.072		
LS	0.106	0.325	73.57	89.67	5.19	14.06	143.40	144.20	144.60	140.30	0.128	11.45	177.6
SI	0.309	0.345	149.80	135.70	1.00	4.50	55.30	55.75	55.82	55.27	0.112	8.55	77.8
RLW	0.113	0.340	76.95	92.76	5.86	15.46	156.30	157.10	157.60	153.00	0.137	13.00	203.8
DWA	0.107	0.325	74.06	90.61	5.09	13.99	142.30	143.00	143.40	139.30	0.125	11.18	175.3
UW	0.386	0.425	166.20	155.80	1.06	4.99	66.40	66.78	66.80	66.24	0.122	10.09	108.0
MGDA-UB	0.217	0.368	126.80	104.60	3.22	5.69	88.37	89.40	89.32	88.01	0.120	11.64	120.5
PCGrad	0.106	0.293	75.85	88.33	3.94	9.15	116.36	116.80	117.20	114.50	0.110	9.55	125.7
CAGrad	0.118	0.321	83.51	94.81	3.21	6.93	113.99	114.30	114.50	112.30	0.116	10.73	112.8
IMTL-G	0.136	0.287	98.31	93.96	1.75	5.69	101.40	102.40	102.00	100.10	0.096	9.27	77.2
Nash-MTL	0.102	0.248	82.95	81.89	2.42	5.38	74.50	75.02	75.10	74.16	0.093	6.64	62.0
FAMO	0.150	0.300	94.00	95.20	1.63	4.95	70.82	71.20	71.20	70.30	0.100	8.09	58.5
FairGrad	0.117	0.253	87.57	84.00	2.15	5.07	70.89	71.17	71.21	70.88	0.095	7.00	57.9
ConsMTL	0.115	**0.202**	82.69	**67.58**	1.61	**3.33**	48.84	49.04	49.07	49.63	0.077	3.27	23.2
GO4Align	0.170	0.350	102.40	119.00	1.22	4.94	53.90	54.30	54.30	53.90	0.110	7.73	52.7
BiLB4MTL ($\tau = \sigma$)	0.230	0.290	123.89	111.95	**0.97**	3.99	42.73	43.10	43.20	43.10	0.097	5.82	49.5
RGB	**0.081**	0.205	**70.30**	70.31	1.60	3.74	**39.27**	**39.56**	**39.55**	**39.55**	**0.075**	**1.64**	**12.4**

Table 7: Detailed standard error on QM9. All metrics are MAE (lower is better). Results are averaged over three random seeds.

Method	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\langle R^2 \rangle$	ZPVE	U_0	U	H	G	C_v	$\Delta m\%$ ↓
	MAE ↓											
RGB	0.081	0.205	70.30	70.31	1.60	3.74	39.27	39.56	39.55	39.55	0.075	12.4
± stderr	±0.003	±0.008	±2.01	±3.56	±0.07	±0.18	±1.37	±1.45	±1.37	±1.15	±0.003	±3.42

Table 8: Detailed standard error on NYUv2. Hierarchical headers. Lower is better for ↓, higher for ↑. Results are averaged over three random seeds.

Method	Segmentation		Depth		Surface Normal					$\Delta m\%$ ↓
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Angle Dist. ↓		Within t° ↑			
					Mean	Median	11.25	22.5	30	
RGB	44.25	68.84	0.5125	0.2126	24.47	18.80	30.07	57.70	70.12	-8.61
± stderr	±0.38	±0.41	±0.0093	±0.0056	±0.10	±0.15	±0.38	±0.28	±0.18	±0.47

Table 9: Detailed standard error on Cityscapes (3 tasks). Lower is better for ↓, higher for ↑. Results are averaged over three random seeds.

Method	mIoU [%] ↑	L1 [px] ↓	MSE ↓	$\Delta m\%$ ↓
RGB	65.72	10.06	0.324	-1.70
± stderr	±0.09	±0.0354	±0.0012	±0.26

Table 10: Detailed standard error on CelebA (40 tasks), measured by $\Delta m\%$ (lower is better). Results are averaged over three random seeds.

Method	$\Delta m\% \downarrow$
RGB	-1.49
$\pm$ stderr	$\pm$0.37

Table 11: Comparison of our proposed RGB method against the RotoGrad baseline on the NYUv2 dataset using a SegNet architecture. Single-task learning (STL) results are sourced directly from the RotoGrad Javaloy & Valera (2022). Since the experiment setup is missing from RotoGrad repository, thus we follow the setup Navon et al. (2022) and align the identical architecture of SegNet Javaloy & Valera (2022) to serve as a reference for rough comparison.

	Segmentation ↑		Depth ↓		Normal ↓		Within t° ↑			
Method	mIoU	PixAcc	Abs	Rel	Mean	Med.	11.25	22.5	30	$\Delta m\% \downarrow$
STL	39.21	64.59	0.70	0.27	25.09	19.18	30.01	57.33	69.30	0.00
RotoGrad	39.32	66.07	0.53	0.21	26.01	20.80	27.18	54.02	66.53	-1.97
RGB (Ours)	**43.86**	**68.27**	**0.51**	**0.20**	**24.40**	**18.67**	**31.17**	**58.51**	**70.57**	**-9.31**

Q Ablation Study: Original vs. Kernel-Based RGB

Table 12: Comparison of Performance ($\Delta m\%$), Training Time, and Memory Usage between Original and Kernel-based RGB methods across NYUv2, QM9, and Cityscapes.

Method	NYUv2			QM9			Cityscapes		
	$\Delta m\% \downarrow$	Time (mins)	Memory (GB)	$\Delta m\% \downarrow$	Time (mins)	Memory (GB)	$\Delta m\% \downarrow$	Time (mins)	Memory (GB)
RGB (Original)	-5.92	20.3	9.1	17.5	6.5	2.8	**-1.81**	10.8	8.51
RGB (Kernel)	**-8.61**	8.0	9.1	**12.4**	6.6	2.8	-1.70	5.1	7.98

This section is to analyze the transition from the Original RGB method to the Kernel-based RGB variant. While both methods aim to align task gradients to minimize conflict through a rotation-based optimization, the kernel-based approach significantly streamlines the computational bottleneck, specifically the iterative optimization of the rotation angle α. This refinement results in a dramatic increase in training efficiency while maintaining the performance gains of the original algorithm.

Table 12 compares the Original RGB method with its Kernel-based counterpart under identical settings, including the same rotation objective, parameterization, and α optimization steps ($\alpha_{\text{steps}} = 50$). The only difference between the two methods lies in how task similarity and orientation losses are computed during the internal α optimization loop.

Across all three datasets, the Kernel-based RGB achieves a substantial reduction in training time, with over a 2× speedup on NYUv2 (20.3 to 8.0 minutes) and Cityscapes (10.8 to 5.1 minutes), while maintaining comparable or improved performance. Notably, the kernel variant even improves $\Delta m\%$ on NYUv2 (-8.61 vs -5.92). Memory consumption remains unchanged or slightly reduced, indicating that the efficiency gains are not achieved at the cost of additional hardware resources.

The speedup stems from eliminating repeated high-dimensional tensor operations inside the α optimization loop. In the Original RGB implementation, the rotated gradient matrix $R \in \mathbb{R}^{T\times D}$ is explicitly reconstructed at every step, followed by the computation of $RR^\top$, resulting in a per-iteration cost proportional to $O(T^2D)$. When the number of shared parameters D is large, this repeated computation becomes a major bottleneck.

In contrast, the Kernel-based RGB reformulates the similarity computation using pre-computed inner products between task gradients and their orthogonal components, yielding small $T \times T$ kernel

matrices. As a result, the expensive $O(T^2D)$ computation is performed only once, and each subsequent α update operates purely in task space with complexity $O(T^2)$. This reduces the overall complexity from $O(\alpha_{\text{steps}} \cdot T^2D)$ to $O(T^2D + \alpha_{\text{steps}} \cdot T^2)$.

Importantly, this kernel formulation is mathematically exact as it computes the same cosine similarities and alignment scores as the original method, without approximation. By avoiding repeated operations on large parameter tensors, the kernel variant reduces numerical overhead during α optimization. This reformulation preserves the intended behavior of RGB while making the optimization substantially more efficient, and may contribute to the improved stability observed on certain datasets.

R Computational Complexity and Empirical Overhead Analysis

In this section, we provide a comprehensive theoretical and empirical analysis of time complexity, wall-clock training speed, and memory consumption to evaluate the computational overhead of the inner optimization loop and the scalability of Rotation-Based Gradient Balancing (RGB). To ensure a robust evaluation across diverse domains, all measurements reported in this section were conducted on four major multi-task learning benchmarks, namely CelebA dataset (40 tasks), Cityscapes (3 tasks), NYU (3 tasks), and QM9 (11 tasks).

Space Complexity. RGB requires storing T task gradients $\nabla f_i(\theta)$, their corresponding orthogonal reference vectors w_i, and the gram matrices, resulting in a space complexity of $\mathcal{O}(TD+T^2)$, where D is the number of shared parameters. This is consistent with standard gradient manipulation methods like PCGrad (Yu et al., 2020) and MGDA (Sener & Koltun, 2018), which also require storing individual task gradients. Figure 14a illustrates that RGB (all steps) maintains a peak memory of **4.8 GB**. While this is higher than simple weighting methods like RLW (1.8 GB) and standard manipulation methods like PCGrad (3.4 GB), it remains significantly more efficient than high-overhead methods such as IMTL-G (6.4 GB). On Cityscapes (Figure 14c), RGB maintains a peak memory of **7.98 GB**. While this is higher than simple weighting methods like RLW or DWA (6.99 GB), it remains more efficient than GradNorm (8.63 GB). As shown in Figure 15a, RGB utilizes **9.1 GB** of peak memory, which is on par with FairGrad and GradDrop (9.1 GB) and more efficient than high-overhead methods such as MoCo (9.4 GB). On the molecular benchmark (Figure 15c), RGB consumes **2.8 GB** of peak memory, similar to most of the baselines.

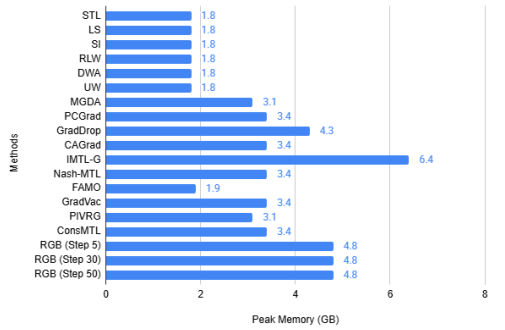
Time Complexity. The primary computational overhead is the inner optimization of the rotation angles $\boldsymbol{\alpha}$. The construction of the interaction matrices A, B, C, E (representing pairwise similarities between gradients and their orthogonal counterparts) scales as $\mathcal{O}(T^2D)$. Once these matrices are computed, the iterative update of α involves $T \times T$ matrix operations. With the inner loop running for α_{steps} iterations, the complexity per training step is $\mathcal{O}(\alpha_{\text{steps}} \cdot T^2)$. Since $T \ll D$, the quadratic scaling with respect to tasks is dominated by the initial gradient projection, making the method highly scalable for large-scale backbones.

Training Time Per Epoch. We compare the wall-clock training time per epoch (gradient computation + manipulation) in Figure 14b, 14d, 15b, and 15d. As expected, simple weighting methods (RLW, DWA, UW) are the fastest (2.6 minutes per epoch) as they involve only scalar multiplication. Among gradient manipulation methods, RGB demonstrates a computational footprint comparable to existing baselines. Specifically, RGB (Step-50) requires approximately **15.9 minutes per epoch** on CelebA, which is significantly faster than MGDA (20.2 mins), GradVac (20.9 mins), Nash-MTL (28.5 mins), and PCGrad (29.9 mins). On Cityscapes, RGB (Step 5) requires **4.3 minutes per epoch**, which is identical to PCGrad (4.3 mins) and faster than GradNorm (6.2 mins). Even with 50 inner optimization steps, the time only increases to 5.1 minutes. On NYU, RGB (Step 5) completes an epoch in **7.2 minutes**, comparable to other manipulation baselines like MoCo (6.5 mins) and Nash-MTL (6.3 mins). On QM9, RGB (Step 5) takes **4.6 minutes per epoch**, which is nearly identical to GradDrop (4.3 mins).

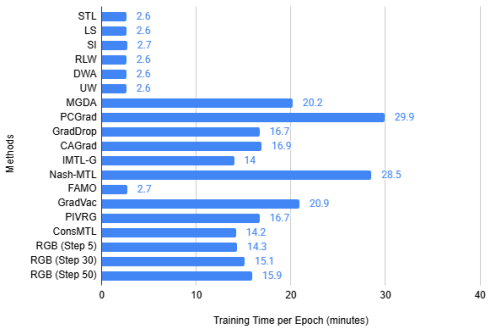
Scalability Conclusion. RGB confines gradient rotation and angle optimization to training. Constructing the Gram matrices costs $\mathcal{O}(T^2D)$, after which each inner iteration costs $\mathcal{O}(T^2)$. With α_{steps} inner iterations, the gradient-manipulation cost per training step is therefore $\mathcal{O}(T^2D + \alpha_{\text{steps}}T^2)$, excluding task-gradient computation. The kernel formulation avoids repeated D-dimensional vector

operations during the inner loop, enabling joint optimization of the task-specific rotation angles with practical training overhead.

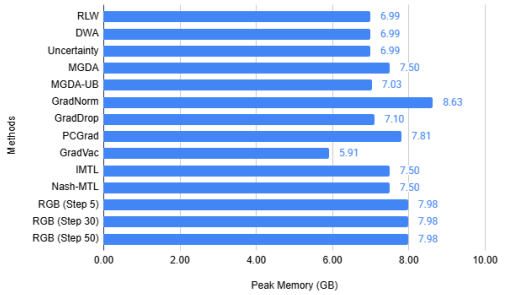
At inference, the model retains the underlying network architecture and uses its learned shared and task-specific parameters for forward propagation. Gradient computation, rotation, and inner optimization are no longer required. RGB therefore introduces no additional inference-time operations or model parameters relative to the same architecture trained without RGB. This ensures that it remains highly efficient for deployment and real world application. In summary, the empirical runtime and memory measurements demonstrate the practical cost of incorporating the alignment–fidelity objective during training.



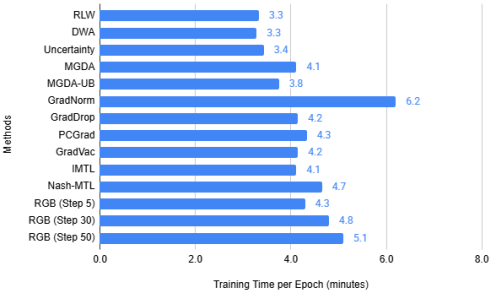
(a) Peak memory usage (GB) (CelebA)



(b) Training time per epoch (minutes) (CelebA)

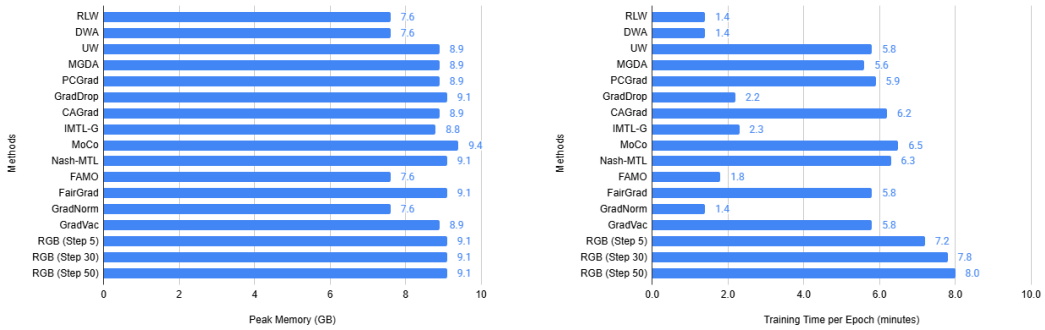


(c) Peak memory usage (GB) (Cityscapes)



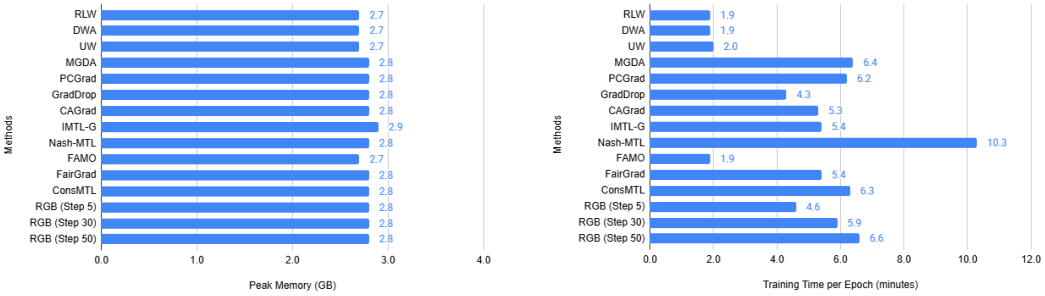
(d) Training time per epoch (minutes) (Cityscapes)

Figure 14: Computational overhead comparison and peak memory usage on CelebA (40 tasks) and Cityscapes (3 tasks).



(a) Peak memory usage (GB) (NYU)

(b) Training time per epoch (minutes) (NYU)



(c) Peak memory usage (GB) (QM9)

(d) Training time per epoch (minutes) (QM9)

Figure 15: Computational overhead comparison and peak memory usage on NYU (3 tasks) and QM9 (11 tasks).