Balanced Multi-Task Learning from an Optimality-Gap Perspective

Ce Liu[1,3] **Jianing Huang**[1,3] **Sicheng Huang**[1,3] **Shu Liu**[1,3] **Xinyu Huang**[2,3] **Hao Yang**[1,3]
[1] Bosch Research Asia Pacific [2] Bosch Research North America
[3] Bosch Center for Artificial Intelligence
{ce.liu, jianing.huang, kevin.yang}@cn.bosch.com

Abstract

Multi-task learning requires a shared update that makes balanced progress across multiple objectives. A key obstacle is that progress is difficult to compare across tasks with different loss scales, units, and local optimization dynamics. We study this problem from an optimality-gap perspective. For each task, we compare the improvement induced by the shared update with the best one-step improvement achievable by optimizing that task in isolation. This comparison yields a scale-invariant normalized improvement rate, which measures how well the shared update serves each task relative to its own local improvement bound. We formulate balanced multi-task optimization as a max-min problem over the normalized improvement rates, thereby prioritizing the worst-served task. A local quadratic approximation leads to the Karush-Kuhn-Tucker (KKT) optimality conditions, showing that the optimal shared update is determined by a small set of bottleneck tasks. Under an isotropic Hessian approximation, the update admits a closed-form expression that depends only on task gradients and a regularized Gram matrix, while the bottleneck set is identified by an active-set procedure. Experiments on diverse benchmarks show improved task balance and competitive performance.

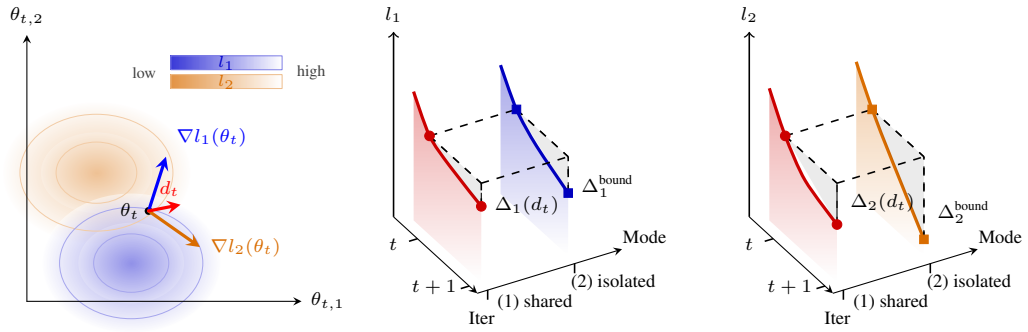
1 Introduction

Multi-task learning (MTL) aims to train a single model that solves multiple tasks simultaneously [13, 90, 80, 20, 68, 62]. It has been widely used in applications such as scene understanding [87, 59, 52], natural language translation [22, 15], and facial attribute recognition [91, 66]. By sharing representations across tasks, MTL can improve data efficiency and generalization.

However, the key challenge is that all tasks must be optimized through a single shared parameter update, even though their objectives are generally not aligned [78, 89, 76]. This shared-update constraint makes task balance a fundamental issue. A descent direction that substantially improves one task may be suboptimal, or even detrimental, for another [88]. Therefore, recent MTL methods seek shared updates that trade off progress across tasks. Existing approaches can be broadly divided into two categories: *i*) loss-oriented methods adjust task weights using homoscedastic uncertainty [37], loss-decay rates [49, 46], or task-level performance signals [29, 64]; and *ii*) gradient-oriented methods operate directly on task gradients, for example, by removing conflicting components [88] or by solving a multi-objective gradient-combination problem [70, 47, 16, 21]. Despite these differences, both families rely on an implicit or explicit measure of how well the current shared update serves each task.

Defining such a measure is nontrivial. Raw loss values are often incomparable across tasks since different tasks may have different loss scales, units, and statistical meanings. Loss reductions are also imperfect indicators of task progress. A task that is already close to a good local optimum may show only a small additional decrease, but this does not imply that it is being neglected by the shared

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Conflicting Gradients (b) Small Gap for Task 1 (c) Large Gap for Task 2

Figure 1: Intuition for the proposed optimality-gap perspective. **(a)** At iteration t, the task gradients can point in different directions, so no single shared update is individually optimal for every task. **(b)** For task 1, the shared step achieves nearly the same loss decrease, $\Delta_1(d_t)$, as the best one-step decrease obtainable when task 1 is optimized alone, $\Delta_1^{\texttt{bound}}$, so the gap is small. **(c)** For task 2, the shared step still reduces the loss, but $\Delta_2(d_t)$ is much smaller than the corresponding isolated-task bound, $\Delta_2^{\texttt{bound}}$, so the gap is large.

update. Conversely, a task with a large numerical loss decrease may still be poorly served relative to its own achievable improvement. Balancing tasks according to raw losses or loss-decay rates can introduce scale-dependent bias and may obscure the actual remaining room for improvement.

In this paper, we address this issue from an optimality-gap perspective. In mathematical optimization, gap-based quantities, such as optimality gaps, integrality gaps, and duality gaps, measure the distance between a feasible solution and an appropriate benchmark or bound [85, 6, 31]. We introduce this principle to MTL by evaluating each shared update relative to the best one-step improvement that the same task could achieve if optimized in isolation. This comparison provides a task-specific and scale-invariant notion of progress. A task with a small gap is already well served by the shared update, whereas a task with a large gap remains underserved. Figure 1 illustrates why raw loss reduction can be misleading under this view. To estimate the single-task benchmark, we relax the coupling induced by the shared update and allow each task to define its own individualized improvement bound.

Building on this perspective, we develop a practical optimization method for balanced shared updates. We first define a normalized improvement rate using the task-wise optimality gap and derive a tractable approximation based on a local quadratic model. We then formulate balanced MTL as a max-min problem over the normalized improvement rates, which prioritizes the worst-served task. The resulting problem admits a characterization through the Karush-Kuhn-Tucker (KKT) optimality conditions [41, 9]. This characterization shows that the optimal shared update is governed by a subset of bottleneck tasks. Under an isotropic Hessian approximation, the update further reduces to a closed-form expression involving only task gradients and a small regularized Gram-matrix system. An active-set procedure is used to identify the bottleneck tasks. Experiments on NYU-v2 [73], CityScapes [19], QM9 [7], and CelebA [50] show that the proposed method improves task balance and achieves competitive performance. Code is available at `https://github.com/cnexah/OG-MTL`.

2 Related Work

Multi-Task Learning. Multi-task learning (MTL) has been studied from many perspectives, including task grouping [78, 89, 76, 72, 27], network architecture design [54, 40, 51, 10, 69, 11, 28, 30, 77, 2, 84], and other directions [14, 24, 34, 44, 35]. As our contribution is an optimization method for balancing shared updates, we focus on optimization-based task balancing in this section.

Optimization-based MTL methods commonly balance tasks either through loss-oriented methods or gradient-oriented methods. Loss-oriented methods adjust task weights using uncertainty [37], random weighting [43], task priority [29], self-paced scheduling [57], relative training progress [46], smooth Tchebycheff scalarization [45], fair resource allocation [4], performance-variance reduction [64], or task-specific parameter learning [65]. These methods infer task status from losses, performance metrics, or fairness criteria. Gradient-oriented methods instead form the shared update

from task gradients, for example through Pareto-stationary minimum-norm combinations [70, 21], gradient normalization [17], conflicting gradient projection or sign dropout [88, 18], conflict-averse or direction-oriented updates [47, 86], bargaining-based optimization [58], independent component alignment [71], Bayesian uncertainty-based coordinate-wise gradient aggregation [1], sharpness-aware perturbations [5, 63], Shapley-fair gradient aggregation [82], or joint loss-gradient balancing with maximum-norm gradient equalization [42]. Our method also uses gradient information, but balances a different quantity: each task's shared improvement relative to its own isolated one-step improvement bound, yielding a unit-invariant optimality-gap criterion. Moreover, the normalization arises from our Hessian approximation and must be applied consistently in both coefficient optimization and the final update to preserve the aggregation objective.

Optimality Gap. Gap-based quantities are widely used in optimization to assess solution quality relative to a bound or reference solution [85, 53, 6, 39, 31]. Related concepts also appear in machine learning, such as duality gaps in neural-network and constrained reinforcement learning [23, 83, 61], gap-dependent regret and sub-optimality bounds in bandits and reinforcement learning [75, 74, 33, 81], gap-aware learning-rate scheduling for adversarial nets [32], stochastic-instance complexity for Hedge [56], risk-score optimization [79], and evaluation of learned optimization bounds [12]. In this paper, we use the optimality gap to compare task-wise progress within a single shared MTL update.

3 Method

A shared-update rule is developed from task-wise optimality gaps. In this section, we first define this gap-based measure and derive a quadratic approximation of the normalized improvement rate. We then formulate multi-task optimization as a max-min problem and derive an efficient active-set solver under an isotropic Hessian approximation.

3.1 Problem Setup

We consider the standard multi-task learning setting with a single shared model, where the key challenge is finding a common update direction for potentially misaligned task gradients.

Multi-Task Objective. Consider K tasks optimized using a model parameterized by $\theta \in \mathbb{R}^m$. Each task $i \in [K]$ is associated with a loss function $l_i(\theta) : \mathbb{R}^m \to \mathbb{R}$. A common formulation minimizes a weighted sum of task losses, $\min_\theta \sum_{i=1}^K w_i l_i(\theta)$, where $w_i \geq 0$ denotes the weight of task i. At training iteration t, let $g_i \coloneqq \nabla l_i(\theta_t)$ denote the gradient of task i at the current parameters θ_t. A shared-gradient method forms a common gradient $d_t = \sum_{i=1}^K w_i g_i$. For analysis, we use the simplified update $\theta_{t+1} = \theta_t - \alpha d_t$ with step size $\alpha > 0$, which may differ from the optimizer update used in practice.

Gradient Conflicts. The central difficulty is that task gradients are not necessarily aligned. A direction that induces large descent for one task may provide little benefit, be suboptimal, or even increase the loss of another task. For task j, a first-order approximation gives

$$l_j(\theta_{t+1}) - l_j(\theta_t) \approx -\alpha g_j^\intercal d_t. \tag{1}$$

Thus, when $g_j^\intercal d_t < 0$, the shared update is locally an ascent direction for task j, leading to an increase in the loss [46, 47, 88]. This conflict motivates the need for a shared direction that explicitly balances task-wise improvement, rather than relying on task weights or signals that are sensitive to loss scales.

3.2 Optimality Gap

A balanced shared direction should serve all tasks in a comparable way. Raw losses and raw loss reductions are inadequate for this purpose. Different tasks may have losses with different scales, units, and statistical meanings. Moreover, a task near a good local optimum may exhibit only a small additional decrease, even when the shared update already serves it well. We therefore measure task progress relative to a task-specific one-step benchmark.

Optimality gaps are standard measures of solution quality in mathematical optimization, where a feasible solution is compared with an appropriate bound or reference value [6]. Motivated by this idea, we define a local optimality-gap measure for multi-task optimization. For task i, let the improvement

induced by the shared direction d_t be

$$\Delta_i(d_t) \coloneqq l_i(\theta_t) - l_i(\theta_t - \alpha d_t). \tag{2}$$

We compare this quantity with the best one-step improvement that task i could achieve if optimized in isolation. Specifically, let

$$\Delta_i^{\texttt{bound}} \coloneqq \max_{u \in \mathcal{U}_t} \left[l_i(\theta_t) - l_i(\theta_t - u) \right]. \tag{3}$$

Here, $\mathcal{U}_t \subseteq \mathbb{R}^m$ is the set of admissible one-step updates at iteration t. We assume that the maximum defining $\Delta_i^{\texttt{bound}}$ is attained and strictly positive for each task considered. The optimality gap of the shared direction for task i can then be written as

$$\mathrm{Gap}_i(d_t) = \frac{\Delta_i^{\texttt{bound}} - \Delta_i(d_t)}{\Delta_i(d_t)} = \frac{\Delta_i^{\texttt{bound}}}{\Delta_i(d_t)} - 1, \tag{4}$$

whenever $\Delta_i(d_t) > 0$. Since the additive constant does not affect the comparison between shared directions, we use the equivalent normalized improvement rate:

$$r_i(d_t) \coloneqq \frac{\Delta_i(d_t)}{\Delta_i^{\texttt{bound}}}. \tag{5}$$

This quantity satisfies $r_i(d_t) = 1/(1 + \mathrm{Gap}_i(d_t))$ when the gap is defined. It is invariant to affine rescaling of the task loss and therefore provides a comparable measure of progress across heterogeneous tasks. A value of $r_i(d_t) = 1$ indicates that the shared direction matches the isolated one-step benchmark for task i. A smaller positive value means that the task is only partially served by the shared update, while $r_i(d_t) < 0$ indicates that the shared update locally increases the task loss.

The normalized improvement rate has the following three desirable properties.

Properties of the normalized improvement rate

P1. Affine invariance. For any transformed loss $\tilde{l}_i = a_i l_i + b_i$ with $a_i > 0$,

$$\tilde{r}_i(d_t) = r_i(d_t). \tag{6}$$

P2. Threshold equivalence. For any $c \in [0, 1]$ and any task with $\Delta_i^{\texttt{bound}} > 0$,

$$r_i(d_t) \geq c \quad \Longleftrightarrow \quad \Delta_i(d_t) \geq c\, \Delta_i^{\texttt{bound}}. \tag{7}$$

P3. Monotonicity in the isolated bound. For fixed $\Delta_i(d_t) > 0$,

$$\frac{\partial r_i(d_t)}{\partial \Delta_i^{\texttt{bound}}} = -\frac{\Delta_i(d_t)}{(\Delta_i^{\texttt{bound}})^2} < 0. \tag{8}$$

These properties make the normalized improvement rate more suitable for task balancing than raw loss-based signals. P1 makes it scale-agnostic by eliminating arbitrary differences in loss units and magnitudes. P2 gives a threshold c a uniform interpretation across tasks: the shared update achieves at least a c-fraction of the task's isolated one-step improvement potential. P3 accounts for local task difficulty by requiring proportionally larger shared improvement when the isolated improvement potential is larger. Together, these properties make $r_i(d_t)$ a calibrated measure of relative progress across heterogeneous tasks. This view is also consistent with standard optimization practice, where solution quality is often assessed through objective residuals, duality gaps, or primal–dual optimality gaps, rather than raw objective values alone [9, 6, 39, 36].

3.3 Quadratic Approximation

The exact normalized improvement rate in Eq. (5) is generally intractable, since the isolated benchmark requires solving a task-specific inner optimization problem. We therefore approximate it using a local quadratic model. Let $H_i \coloneqq \nabla^2 l_i(\theta_t)$ denote the Hessian of task i at the current iterate. Around θ_t, the loss after the shared update can be approximated as

$$l_i(\theta_t - \alpha d_t) \approx l_i(\theta_t) - \alpha g_i^\intercal d_t + \frac{\alpha^2}{2} d_t^\intercal H_i d_t. \tag{9}$$

Thus, the predicted decrease of task i under the shared direction is

$$\widehat{\Delta}_i(d_t) = \alpha g_i^\intercal d_t - \frac{\alpha^2}{2} d_t^\intercal H_i d_t. \tag{10}$$

Under the same quadratic surrogate, the isolated benchmark is obtained by allowing task i to optimize its own local step independently. Assuming H_i is positive definite, or replaced by a positive-definite local curvature approximation, the optimal isolated step is the Newton step $H_i^{-1} g_i$. Substituting this step into the quadratic model gives the predicted isolated decrease

$$\beta_i \coloneqq \widehat{\Delta}_i^{\texttt{bound}} = \frac{1}{2} g_i^\intercal H_i^{-1} g_i. \tag{11}$$

The quadratic approximation of the normalized improvement rate is therefore

$$r_i(d_t) \approx \widehat{r}_i(d_t) = \frac{\alpha g_i^\intercal d_t - \frac{\alpha^2}{2} d_t^\intercal H_i d_t}{\beta_i}. \tag{12}$$

Here, β_i serves as a task-specific normalization factor. It captures the local improvement potential of task i through the gradient–curvature geometry $g_i^\intercal H_i^{-1} g_i$, and converts the raw predicted decrease into a comparable relative progress measure.

3.4 Max-Min Gap Balancing

Since the normalized improvement rate is a monotone transformation of the optimality gap whenever the gap is well defined, minimizing the worst gap is equivalent to maximizing the worst normalized improvement rate. We therefore choose the shared direction by solving

$$\max_{d_t} \min_{i \in [K]} r_i(d_t). \tag{13}$$

This objective favors a direction whose least-served task still receives the largest possible relative improvement. Introducing an auxiliary variable τ, the problem can be equivalently written as

$$\min_{\tau,\, d_t} \quad -\tau \quad \text{subject to} \quad r_i(d_t) \geq \tau, \qquad \forall i \in [K]. \tag{14}$$

Here, τ represents the worst task-wise normalized improvement achieved by the shared direction. Under the local quadratic approximation in Eq. (12), the constraints are determined by concave quadratic functions of d_t when H_i is positive semi-definite and $\beta_i > 0$.

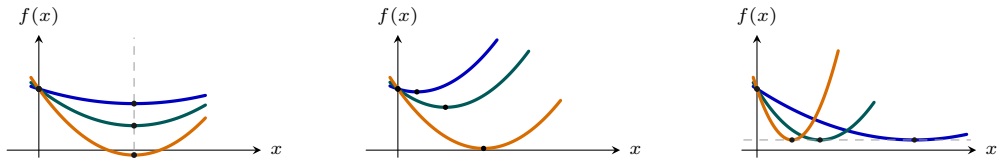
We apply the Karush-Kuhn-Tucker (KKT) conditions to the quadratic surrogate of Eq. (14), replacing r_i by $\widehat{r}_i$. The corresponding saddle-point problem is

$$\max_{\lambda \geq 0} \min_{d_t,\, \tau} \mathcal{L}(d_t, \tau, \lambda), \tag{15}$$

where $\mathcal{L}(d_t, \tau, \lambda) = -\tau + \sum_{i=1}^K \lambda_i \big(\tau - \widehat{r}_i(d_t)\big)$ is the Lagrangian, and $\lambda_i \geq 0$ is the multiplier associated with the constraint $\widehat{r}_i(d_t) \geq \tau$. The KKT conditions are

$$\lambda_i \geq 0, \ \forall i \in [K], \tag{16a}$$
$$\widehat{r}_i(d_t) \geq \tau, \ \forall i \in [K], \tag{16b}$$
$$\lambda_i \big(\widehat{r}_i(d_t) - \tau\big) = 0, \ \forall i \in [K], \tag{16c}$$
$$\sum\nolimits_{i=1}^K \lambda_i = 1, \tag{16d}$$
$$d_t - \left(\sum\nolimits_{i=1}^K \frac{\lambda_i \alpha}{\beta_i} H_i\right)^{-1} \left(\sum\nolimits_{i=1}^K \frac{\lambda_i}{\beta_i} g_i\right) = 0. \tag{16e}$$

Equations (16a) and (16b) express dual and primal feasibility. Equations (16d) and (16e) follow from stationarity; the latter assumes that the weighted Hessian is invertible. Complementary slackness in Eq. (16c) implies that tasks with $\lambda_i > 0$ attain the bottleneck rate, $\widehat{r}_i(d_t) = \tau$. Tasks with zero multipliers satisfy $\widehat{r}_i(d_t) \geq \tau$ and may also be binding. Thus, the support of the multipliers need not include every binding constraint.



(a) $\gamma_i = |f'(0)|$ (b) $\gamma_i = 1$ (c) $\gamma_i = |f'(0)|^2$

Figure 2: One-dimensional quadratic surrogates induced by different choices of γ_i. Each panel shows three task-specific quadratics that pass through the same point at $x = 0$ but have different slopes there, illustrating how the curvature scale changes the implied single-task optima. **(a)** $\gamma_i = |f'(0)|$: all three surrogates share a common minimizer. **(b)** $\gamma_i = 1$: the surrogates have different minimizers and different minimum values. **(c)** $\gamma_i = |f'(0)|^2$: the surrogates share the same minimum value; the dashed horizontal line marks this common level.

3.5 Isotropic Hessian and Active-Set Solution

Isotropic Hessian. Solving the KKT system in Eq. (16) with full task Hessians H_i is prohibitively expensive for modern neural networks. We therefore adopt an isotropic curvature surrogate $H_i \approx \gamma_i I$, where I is the identity matrix and $\gamma_i > 0$ is a task-specific curvature scale. Following the practice of using gradient magnitude as a local optimization scale [55, 38], we set $\gamma_i = \|g_i\|$. Under this isotropic quadratic model, the isolated step for task i is g_i/γ_i, whose length is $\|g_i\|/\gamma_i$. The choice $\gamma_i = \|g_i\|$ therefore assigns each task a unit-length isolated benchmark. This prevents the normalization from favoring tasks solely because they have larger gradients or smaller surrogate curvature. It also simplifies the isolated improvement bound to $\beta_i = \frac{1}{2} g_i^\intercal (\gamma_i I)^{-1} g_i = \frac{\|g_i\|}{2}$. Figure 2 contrasts this choice with alternative curvature scalings, such as $\gamma_i = 1$ and $\gamma_i = \|g_i\|^2$.

Gram Regularization. For convenience, define the normalized task gradients and their Gram matrix:

$$\hat{g}_i \coloneqq \frac{g_i}{2\beta_i}, \qquad G_{ij} \coloneqq \hat{g}_i^\intercal \hat{g}_j. \tag{17}$$

With the isotropic choice above, $\hat{g}_i = g_i/\|g_i\|$. In practice, G is estimated from mini-batch gradients and may be noisy or nearly singular, especially when several normalized gradients are strongly aligned. Since the closed-form multipliers require inverting an active-set submatrix of G, such ill-conditioning can make the update numerically unstable and overly sensitive to batch noise. We therefore use a diagonally regularized Gram matrix $\tilde{G} = G + \rho I$, where $\rho > 0$ controls the regularization strength. Because G is positive semidefinite, $\tilde{G}$ is positive definite and invertible [67]. This regularization stabilizes the active-set solve and yields smoother updates across iterations.

Active-Set Solution. In the isotropic Hessian approximation, the stationarity condition in Eq. (16e) is

$$d_t^*(\lambda) = \frac{1}{\alpha} \sum_{i=1}^{K} \lambda_i \hat{g}_i = \sum_{i=1}^{K} \underbrace{\frac{\lambda_i}{2\alpha\beta_i}}_{=: w_i} g_i. \tag{18}$$

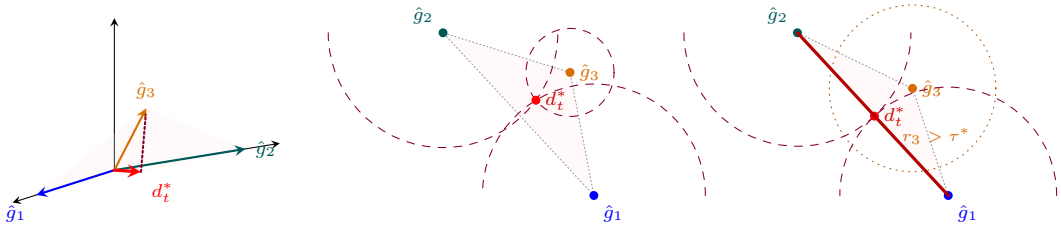
Thus, once the multipliers are known, the shared direction is obtained as a weighted combination of task gradients, with effective task weight $w_i = \lambda_i/(2\alpha\beta_i)$. Since $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$, the scaled direction $\alpha d_t^*(\lambda)$ is a convex combination of normalized gradients, as illustrated in Figure 3(a).

It remains to solve for the multipliers. Consider a nonempty candidate active set $A \subseteq [K]$, with $\lambda_i^* = 0$ for $i \notin A$. Multipliers within A are computed before checking nonnegativity. Let $\tilde{G}_A$ denote the submatrix of $\tilde{G}$ indexed by A, and define $s_A \coloneqq \mathbf{1}_A^\intercal \tilde{G}_A^{-1} \mathbf{1}_A$. The regularized quadratic-program value and the corresponding multipliers restricted to A are

$$\tau_A^* = \frac{1}{s_A}, \qquad \lambda_A^* = \frac{\tilde{G}_A^{-1} \mathbf{1}_A}{s_A}. \tag{19}$$

The shared direction d_t^* is subsequently recovered from Eq. (18). The full derivation is provided in Appendix A. Figures 3(b) and 3(c) show that all tasks are active and one task becomes inactive.

The candidate solution is valid if it satisfies two conditions. First, the active multipliers must be nonnegative, that is, $\lambda_A^* \geq 0$ componentwise. Second, every inactive task must satisfy the regularized



(a) Simplex in Normalized-Gradient Basis (b) All Tasks Active (c) One Inactive Task

Figure 3: Geometry of the optimal shared direction d_t^* in the normalized-gradient simplex. The dashed and dotted curves denote level sets of the task-wise rates $r_i(d_t)$. **(a)** $d_t^* = \frac{1}{\alpha}\sum_i \lambda_i \hat{g}_i$ lies on the simplex face determined by the active multipliers. **(b)** When all tasks are active, d_t^* lies in the interior and equalizes the rates, $r_1 = r_2 = r_3 = \tau^*$. **(c)** When task 3 is inactive, d_t^* lies on the $\lambda_3 = 0$ edge, with $r_1 = r_2 = \tau^*$ and $r_3 > \tau^*$.

Algorithm 1 The proposed optimality-gap balancing method

Require: Current parameter θ_t, task gradients $g_i \neq 0$ for all $i \in [K]$, regularizer $\rho > 0$
for $i = 1$ to K **do**
  Compute $\beta_i \leftarrow \|g_i\|/2$, and $\hat{g}_i \leftarrow g_i/(2\beta_i)$
end for
Form the Gram matrix with entries $G_{ij} \leftarrow \hat{g}_i^\intercal \hat{g}_j$ for all $i, j \in [K]$, and $\tilde{G} \leftarrow G + \rho I$
Initialize the active set $A \leftarrow [K]$ and the visited-set collection $\mathcal{V} \leftarrow \emptyset$
while True **do**
  Add the current active set to the visited collection: $\mathcal{V} \leftarrow \mathcal{V} \cup \{A\}$
  Compute $s_A \leftarrow \mathbf{1}_A^\intercal \tilde{G}_A^{-1} \mathbf{1}_A$, and $\tau_A^* \leftarrow 1/s_A$
  Compute $\lambda_A^* \leftarrow \tilde{G}_A^{-1}\mathbf{1}_A/s_A$, and $\lambda_i^* \leftarrow 0$ for all $i \notin A$
  if $\lambda_A^* \geq 0$ componentwise and $(\tilde{G}\lambda^*)_j \geq \tau_A^*$ for all $j \notin A$ **then break**
  else if there exists active $i \in A$ with $\lambda_i^* < 0$ **then** set $A' \leftarrow A \setminus \{\min\{i \in A : \lambda_i^* < 0\}\}$
  else set $A' \leftarrow A \cup \{\min\{j \notin A : (\tilde{G}\lambda^*)_j < \tau_A^*\}\}$
  if $A' \in \mathcal{V}$ **then** set A' to the lexicographically smallest nonempty subset of $[K]$ not in $\mathcal{V}$
  Set $A \leftarrow A'$
end while
return $\sum_i \lambda_i^* \hat{g}_i$

inactive-set optimality condition $(\tilde{G}\lambda^*)_j \geq \tau_A^*$ for $j \notin A$. In the unregularized case, this condition is equivalent to $\widehat{r}_j(d_t^*) \geq \tau_A^*$. If an active task has a negative multiplier, it is removed from the active set. If an inactive task violates the inactive-set optimality condition, it is added to the active set. Whenever more than one index is eligible to be added or removed, we choose the smallest such index and use a no-repeat anti-cycling check on active sets. The system is then solved again until both validity conditions are satisfied.

These validity conditions justify removing a task from the candidate active set: a negative multiplier violates KKT dual feasibility and cannot define a valid task weight. Removal therefore serves to restore feasibility, rather than heuristically discard a task [60]. At termination, inactive tasks satisfy the applicable optimality condition; in the unregularized case, they already meet or exceed the bottleneck rate, $\widehat{r}_j(d_t^*) \geq \tau_A^*$, despite having zero weight.

Algorithm 1 summarizes the active-set procedure, whose finite termination is proved in Appendix B. It returns the aggregated gradient $\sum_i \lambda_i^* \hat{g}_i = \alpha d_t^*$, which is independent of α. The outer optimizer uses this vector in place of the usual loss gradient, with its learning rate, momentum, and preconditioning determining the actual parameter update.

4 Experiment

Datasets. We evaluate on four standard MTL benchmarks: NYU-v2 [73] (semantic segmentation, depth estimation, and surface normal prediction), CityScapes [19] (semantic segmentation and depth

estimation), QM9 [7] (11 molecular-property prediction tasks), and CelebA [50] (40 facial-attribute classification tasks). For QM9, we follow prior work [58] and use the same split [25, 26].

Evaluation Metrics. Following prior work [58], we report two standard MTL metrics. *i)* $\boldsymbol{\Delta m}$ measures the average per-task performance change of a method m relative to the single-task learning (STL) baseline b: $\boldsymbol{\Delta m}\% = \frac{1}{K}\sum_{k=1}^{K}(-1)^{\delta_k}(M_{m,k} - M_{b,k})/M_{b,k} \times 100$, where $M_{m,k}$ and $M_{b,k}$ denote the values of metric M_k for m and the STL baseline, and $\delta_k = 1$ when higher values are better and $\delta_k = 0$ when lower values are better. *ii)* MR is the mean rank of a method across tasks.

Implementation Details. We perform experiments using PyTorch 1.13.1 and CUDA 12.4 on a single NVIDIA A800 GPU. More details are in Appendix F.

4.1 Performance Comparison with Prior Works

We compare our method with recent gradient-manipulation MTL baselines, including DB-MTL [42], FairGrad [4], BayesAgg-MTL [1], FAMO [46], and NashMTL [58], as well as standard reference methods such as single-task learning (STL), linear scalarization (LS), and scale-invariant optimization (SI). We follow the same experimental settings as prior work [46]. We repeat each experiment three times and report the mean and standard deviation. Results for CityScapes and CelebA are reported in Table 1, results for QM9 are shown in Table 2, and results for NYU-v2 are given in Table 3. Overall, our method is competitive across benchmarks, with relative strengths varying across tasks and evaluation metrics. In particular, on CityScapes it reduces $\boldsymbol{\Delta m}$ from 5.18 to 2.50, and on CelebA from 0.37 to -0.34, demonstrating the effectiveness of the proposed design.

Table 1: Results on CityScapes (2 tasks) and CelebA (40 tasks) datasets.

Method	CityScapes						CelebA	
	Segmentation		Depth		MR↓	$\boldsymbol{\Delta m}\downarrow$	MR↓	$\boldsymbol{\Delta m}\downarrow$
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓				
STL	74.01	93.16	0.0125	27.77				
LS	75.18	93.49	0.0155	46.77	8.00	22.60	6.93	6.28
SI	70.95	91.73	0.0161	33.83	10.50	14.11	8.60	7.83
RLW [43]	74.57	93.41	0.0158	47.79	10.25	24.38	5.98	5.22
DWA [49]	75.24	93.52	0.0160	44.37	7.75	21.45	7.60	6.95
UW [37]	72.02	92.85	0.0140	30.13	7.25	5.89	6.30	5.78
MGDA [70]	68.84	91.54	0.0309	33.50	11.00	44.14	12.03	10.93
PCGrad [88]	75.13	93.48	0.0154	42.07	8.25	18.29	7.58	6.65
CAGrad [47]	75.16	93.48	0.0141	37.60	7.00	11.64	7.10	6.20
IMTL-G [48]	75.33	93.49	0.0135	38.41	5.50	11.10	5.40	4.67
NashMTL [58]	75.41	93.66	**0.0129**	35.02	3.25	6.82	5.65	4.97
FAMO [46]	74.54	93.29	0.0145	32.59	7.75	8.13	5.48	4.72
FairGrad [4]	75.72	**93.68**	0.0134	32.25	2.25	5.18	6.15	0.37
Ours	**75.74**	93.64	0.0131	**29.93**	**1.75**	**2.50**	**5.03**	**-0.34**
(error bar)	(±0.002)	(±0.001)	(±0.001)	(±1.578)		(±0.418)		(±0.521)

4.2 Ablations and Further Analysis

To better understand our approach, we perform ablations and further analysis on CityScapes dataset.

***i)* The effect of β_i.** We first examine whether β_i provides a reliable estimate of $\Delta_i^{\texttt{bound}}$. To this end, we train separate single-task models for semantic segmentation and depth estimation, and for each mini-batch we measure the loss decrease induced by one parameter update. Figure 4 (a) and (b) plots this loss reduction against β_i. In both cases, β_i is positively correlated with the observed decrease, and for semantic segmentation (a) the Pearson correlation coefficient reaches 0.87. We then assess the importance of this normalization by replacing β_i with the constant 1 during multi-task training, which effectively evaluates task improvement using raw loss reduction. The resulting performance is shown in Figure 5. Comparing (1) with (2), mIoU improves slightly, but the depth metrics Abs Err and Rel Err deteriorate substantially, leading to a worse overall $\boldsymbol{\Delta m}$.

***ii)* Different choices of γ_i.** We next compare our default choice with two alternatives, namely $\gamma_i = 1$ and $\gamma_i = \|g_i\|^2$. The results are shown in Figure 5. When $\gamma_i = 1$, comparing (1) and (3), the depth estimation metrics become slightly worse, which also degrades the overall $\boldsymbol{\Delta m}$. When

Table 2: Results on QM9 (11 tasks) dataset.

Method	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\langle R^2 \rangle$	ZPVE	U_0	U	H	G	c_v	**MR**↓	$\boldsymbol{\Delta m}\downarrow$
	MAE ↓												
STL	0.07	0.18	60.6	53.9	0.50	4.53	58.8	64.2	63.8	66.2	0.07		
LS	0.11	0.33	**73.6**	89.7	5.20	14.06	143.4	144.2	144.6	140.3	0.13	10.64	177.6
SI	0.31	0.35	149.8	135.7	**1.00**	**4.51**	**55.3**	**55.8**	**55.8**	**55.3**	0.11	6.27	77.8
RLW [43]	0.11	0.34	76.9	92.8	5.87	15.47	156.3	157.1	157.6	153.0	0.14	12.09	203.8
DWA [49]	0.11	0.33	74.1	90.6	5.09	13.99	142.3	143.0	143.4	139.3	0.13	10.36	175.3
UW [37]	0.39	0.43	166.2	155.8	1.07	4.99	66.4	66.8	66.8	66.2	0.12	8.18	108.0
MGDA [70]	0.22	0.37	126.8	104.6	3.23	5.69	88.4	89.4	89.3	88.0	0.12	10.82	120.5
PCGrad [88]	0.11	0.29	75.9	88.3	3.94	9.15	116.4	116.8	117.2	114.5	0.11	8.82	125.7
CAGrad [47]	0.12	0.32	83.5	94.8	3.22	6.93	114.0	114.3	114.5	112.3	0.12	10.00	112.8
IMTL-G [48]	0.14	0.29	98.3	93.9	1.75	5.70	101.4	102.4	102.0	100.1	0.10	8.82	77.2
NashMTL [58]	0.10	0.25	82.9	81.9	2.43	5.38	74.5	75.0	75.1	74.2	**0.09**	5.27	62.0
FAMO [46]	0.15	0.30	94.0	95.2	1.63	4.95	70.8	71.2	71.2	70.3	0.10	6.45	58.5
FairGrad [4]	0.12	0.25	87.6	84.0	2.15	5.07	70.9	71.2	71.2	70.9	0.10	5.09	57.9
BayesAgg-MTL [1]	0.12	0.28	87.8	90.4	1.78	5.31	63.3	64.9	66.7	81.9	**0.09**	5.18	**53.2**
DB-MTL [42]	0.11	0.26	89.26	86.59	2.43	5.41	60.33	60.78	60.80	60.59	0.10	**4.36**	58.4
Ours	**0.09**	**0.24**	78.4	**75.5**	2.26	5.50	72.4	72.8	72.9	71.8	**0.09**	4.64	54.3
(error bar)	(±0.003)	(±0.001)	(±0.797)	(±0.834)	(±0.038)	(±0.134)	(±2.025)	(±2.103)	(±2.086)	(±1.862)	(±0.001)		(±1.496)

Table 3: Results on NYU-v2 (3 tasks) dataset.

Method	Segmentation		Depth		Surface Normal					**MR**↓	$\boldsymbol{\Delta m}\downarrow$
	mIoU↑	Pix Acc↑	Abs Err↓	Rel Err↓	Angle Dist↓		Within t° ↑				
					Mean	Median	11.5	22.5	30		
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15		
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	11.89	5.59
SI	38.45	64.27	0.5354	0.2201	27.60	23.37	22.53	48.57	62.32	10.67	4.39
RLW [43]	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	14.22	7.78
DWA [49]	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	10.67	3.57
UW [37]	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	10.44	4.05
MGDA [70]	30.47	59.90	0.6070	0.2555	24.88	19.45	29.18	**56.88**	**69.36**	7.67	1.38
PCGrad [88]	38.06	64.64	0.5550	0.2325	27.41	22.80	23.86	49.83	63.14	11.00	3.97
GradDrop [18]	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	10.00	3.58
CAGrad [47]	39.79	65.49	0.5486	0.2250	26.31	21.58	25.61	52.36	65.58	7.44	0.20
IMTL-G [48]	39.35	65.60	0.5426	0.2256	26.02	21.19	26.20	53.13	66.24	6.78	-0.76
NashMTL [58]	40.13	65.93	0.5261	0.2171	25.26	20.08	28.40	55.47	68.15	4.78	-4.04
FAMO [46]	38.88	64.90	0.5474	0.2194	25.06	19.57	29.21	56.61	68.98	5.33	-4.10
FairGrad [4]	39.74	66.01	0.5377	0.2236	**24.84**	19.60	29.26	56.58	69.16	3.33	-4.66
DB-MTL [42]	**41.42**	**66.45**	0.5251	0.2160	25.03	**19.50**	28.72	56.17	68.73	**2.89**	-5.36
Ours	40.17	66.05	**0.5143**	**0.2100**	25.08	19.65	**29.25**	56.43	68.83	**2.89**	**-5.49**
(error bar)	(±0.011)	(±0.004)	(±0.013)	(±0.006)	(±0.145)	(±0.252)	(±0.005)	(±0.006)	(±0.004)		(±0.742)

$\gamma_i = \|g_i\|^2$, the degradation on depth estimation is much larger, resulting in substantially worse overall performance.

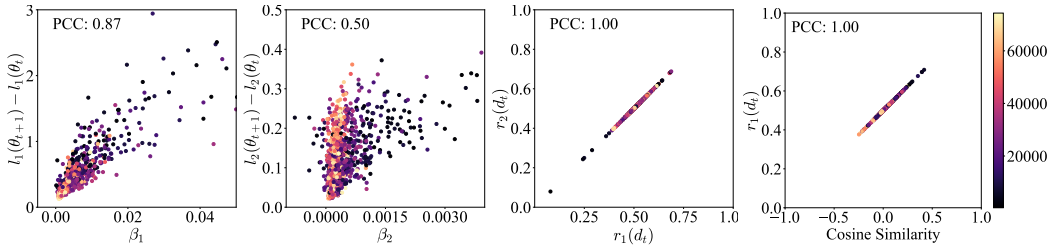
***iii)* The balance among $r_i(d_t)$.** We visualize the solved values of $r_i(d_t)$ throughout training. As shown in Figure 4 (c), the resulting rates remain well balanced over the entire training process. We also observe that, in the two-task setting, $r_i(d_t)$ is strongly correlated with the cosine similarity between the two task gradients, as illustrated in Figure 4 (d). A proof is provided in Appendix D.

4.3 Runtime and Memory Comparison

We compare the computational overhead of different methods in terms of per-batch runtime and memory usage. Specifically, we measure the time required to compute the shared parameter update for each mini-batch and report averages over two training epochs. As shown in Table 4, our method has nearly the same time and memory footprint as NashMTL on both benchmarks. FairGrad is more efficient on CityScapes, which involves only two tasks, but its cost increases substantially on CelebA with 40 tasks. In contrast, our method remains markedly more efficient than FairGrad on CelebA, indicating better scalability as the number of tasks grows.

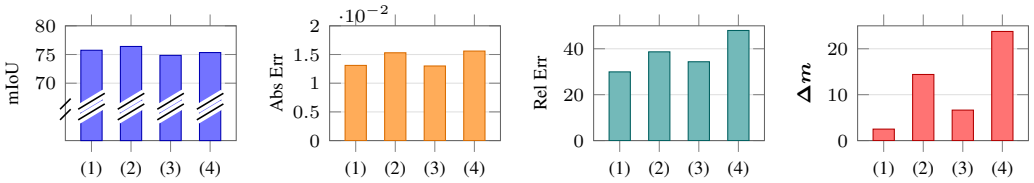
Table 4: Training time (ms) and memory (MB) comparison.

Dataset	Time				Memory			
	NashMTL	FAMO	FairGrad	Ours	NashMTL	FAMO	FairGrad	Ours
CityScapes	610	123	365	608	8908	7506	8736	9102
CelebA	2972	104	3224	2979	4672	3586	5200	4708



(a) Correlation between semantic loss reduction and β_1 (b) Correlation between depth loss reduction and β_2 (c) Balance between $r_2(d_t)$ and $r_1(d_t)$ (d) Correlation between $r_1(d_t)$ and gradient cosine similarity

Figure 4: Ablation analysis of β_i and $r_i(d_t)$ on the CityScapes dataset. **(a)** Correlation between the observed loss reduction and β_1 during single-task training for semantic segmentation. **(b)** Correlation between the observed loss reduction and β_2 during single-task training for depth estimation. **(c)** Relationship between $r_1(d_t)$ and $r_2(d_t)$ during multi-task training, showing that the two rates remain well balanced. **(d)** Relationship between $r_1(d_t)$ and the cosine similarity of the two task gradients during multi-task training. The color bar indicates the training iteration.



(a) Segmentation mIoU (b) Depth Abs Err (c) Depth Rel Err (d) Overall $\boldsymbol{\Delta m}$

Figure 5: Ablation results for different choices of β_i and γ_i on CityScapes. Each panel compares four variants, labeled (1)–(4) on the x-axis, using a different evaluation metric on the y-axis: **(a)** mIoU, **(b)** Abs Err, **(c)** Rel Err, and **(d)** $\boldsymbol{\Delta m}$. Variant (1) is the default method, variant (2) sets $\beta_i = 1$, variant (3) sets $\gamma_i = 1$, and variant (4) sets $\gamma_i = \|g_i\|^2$.

5 Conclusion

We presented a new perspective on multi-task learning based on how well a shared update serves each task relative to that task's own potential for local improvement. This leads to a simple and comparable notion of task improvement that is less sensitive to differences in loss scale across tasks. Across four standard benchmarks, the resulting method delivers more balanced optimization and improved overall performance compared with strong baselines. These results suggest that measuring shared progress relative to task-specific improvement potential is a useful principle for designing shared updates in multi-task optimization.

Limitations and Future Directions. The isotropic Hessian approximation omits direction-dependent curvature and may therefore yield less accurate local models in strongly anisotropic regions. Computing all task gradients, constructing the Gram matrix, and solving the active-set systems also impose computational and memory costs that grow with the number of tasks. Furthermore, the outer convergence guarantee assumes full gradients and an exact, unregularized inner solve; it does not directly apply to the stochastic, regularized implementation used in our experiments. These limitations motivate future work on stable anisotropic curvature surrogates, more scalable solvers, and convergence guarantees that account for stochastic gradients and regularization.

References

[1] Idan Achituve, Idit Diamant, Arnon Netzer, Gal Chechik, and Ethan Fetaya. Bayesian uncertainty for gradient aggregation in multi-task learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 117–134. PMLR, 21–27 Jul 2024.

[2] Ahmed Agiza, Marina Neseem, and Sherief Reda. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition, pages 16196–16205, 2024.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

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

[5] Hao Ban, Gokul Ram Subramani, and Kaiyi Ji. Samo: A lightweight sharpness-aware approach for multi-task optimization with joint global-local perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 785–795, 2025.

[6] Robert E. Bixby, Mary Fenelon, Zonghao Gu, Edward Rothberg, and Roland Wunderling. Mip: Theory and practice—closing the gap. In M. Keyzer, L. Pardo, and M. Vlach, editors, *System Modelling and Optimization: Methods, Theory and Applications*, pages 19–49. Kluwer Academic Publishers, Boston, MA, 2000.

[7] 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.

[8] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

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

[10] Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1385–1394, 2019.

[11] David Brüggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. In *Proceedings of the British Machine Vision Conference*, page 359, 2020.

[12] Quentin Cappart, Emmanuel Goutierre, David Bergman, and Louis-Martin Rousseau. Improving optimization bounds using machine learning: Decision diagrams meet deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1443–1451, 2019.

[13] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[14] Lisha Chen, Heshan Devaka Fernando, Yiming Ying, and Tianyi Chen. Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. In *Advances in Neural Information Processing Systems*, pages 70045–70093, 2023.

[15] Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview. *ACM Computing Surveys*, 56(12):1–32, 2024.

[16] Weiyu Chen, Baijiong Lin, Xiaoyuan Zhang, Xi Lin, Han Zhao, Qingfu Zhang, and James T Kwok. Gradient-based multi-objective deep learning: Algorithms, theories, applications, and beyond. *arXiv preprint arXiv:2501.10945*, 2025.

[17] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the International Conference on Machine Learning*, pages 794–803, 2018.

[18] 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. In *Advances in Neural Information Processing Systems*, pages 2039–2050, 2020.

[19] 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*, pages 3213–3223, 2016.

[20] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

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

[22] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 1723–1732, 2015.

[23] Tolga Ergen and Mert Pilanci. Convex geometry and duality of over-parameterized neural networks. *Journal of Machine Learning Research*, 22(212):1–63, 2021.

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

[25] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *The International Conference on Learning Representations Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[26] Matthias Fey, Jinu Sunil, Akihiro Nitta, Rishi Puri, Manan Shah, Blaž Stojanovič, Ramona Bendias, Barghi Alexandria, Vid Kocijan, Zecheng Zhang, Xinwei He, Jan E. Lenssen, and Jure Leskovec. PyG 2.0: Scalable learning on real world graphs. In *Knowledge Discovery and Data Mining Workshop on Temporal Graph Learning*, 2025.

[27] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, pages 27503–27516, 2021.

[28] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11543–11552, 2020.

[29] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision*, pages 270–287, 2018.

[30] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *Proceedings of the International Conference on Machine Learning*, pages 3854–3863, 2020.

[31] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026.

[32] Hussein Hazimeh and Natalia Ponomareva. Mind the (optimality) gap: A gap-aware learning rate scheduler for adversarial nets. In *Proceedings of The International Conference on Artificial Intelligence and Statistics*, pages 3018–3033, 2023.

[33] Jiafan He, Dongruo Zhou, and Quanquan Gu. Logarithmic regret for reinforcement learning with linear function approximation. In *Proceedings of the International Conference on Machine Learning*, pages 4171–4180, 2021.

[34] Yifei He, Shiji Zhou, Guojun Zhang, Hyokun Yun, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Han Zhao. Robust multi-task learning with excess risks. In *Proceedings of the International Conference on Machine Learning*, pages 18094–18114, 2024.

[35] Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. In *Advances in Neural Information Processing Systems*, pages 48510–48533, 2023.

[36] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the International Conference on Machine Learning*, pages 427–435, 2013.

[37] 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*, pages 7482–7491, 2018.

[38] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[39] Thorsten Koch, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E Bixby, Emilie Danna, Gerald Gamrath, Ambros M Gleixner, Stefan Heinz, et al. Miplib 2010: mixed integer programming library version 5. *Mathematical Programming Computation*, 3(2):103–163, 2011.

[40] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

[41] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1950.

[42] Baijiong Lin, Weisen Jiang, Feiyang Ye, Yu Zhang, Pengguang Chen, Ying-Cong Chen, Shu Liu, Ivor W. Tsang, and James T. Kwok. Dual-balancing for multi-task learning. *Neural Networks*, 195:108317, 2026.

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

[44] Xi Lin, Yilu Liu, Xiaoyuan Zhang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Few for many: Tchebycheff set scalarization for many-objective optimization. In *The International Conference on Learning Representations*, 2025.

[45] Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. In *Proceedings of the International Conference on Machine Learning*, pages 30479–30509, 2024.

[46] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. In *Advances in Neural Information Processing Systems*, pages 57053–57081, 2023.

[47] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, pages 18878–18890, 2021.

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

[49] 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*, pages 1871–1880, 2019.

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

[51] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S. Yu. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems*, page 1593–1602, 2017.

[52] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[53] Wai-Kei Mak, David P Morton, and R Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24(1-2):47–56, 1999.

[54] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[55] Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, et al. A theory on adam instability in large-scale machine learning. *arXiv preprint arXiv:2304.09871*, 2023.

[56] Jaouad Mourtada and Stéphane Gaïffas. On the optimality of the hedge algorithm in the stochastic regime. *Journal of Machine Learning Research*, 20(83):1–28, 2019.

[57] Keerthiram Murugesan and Jaime Carbonell. Self-paced multitask learning with shared knowledge. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2522–2528, 2017.

[58] 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 International Conference on Machine Learning*, pages 16428–16446, 2022.

[59] Duy-Kien Nguyen and Takayuki Okatani. Multi-task learning of hierarchical vision-language representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10501, 2019.

[60] Jorge Nocedal and Stephen J. Wright. *Quadratic Programming*, pages 448–496. Springer New York, New York, NY, 2006.

[61] Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems*, pages 7555–7565, 2019.

[62] Anastasia Pentina and Christoph H. Lampert. Multi-task learning with labeled and unlabeled tasks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2807–2816. PMLR, 06–11 Aug 2017.

[63] Hoang Phan, Lam Tran, Quyen Tran, Ngoc N. Tran, Tuan Truong, Qi Lei, Nhat Ho, Dinh Phung, and Trung Le. Beyond losses reweighting: Empowering multi-task learning via the generalization perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2440–2450, 2025.

[64] Xiaohan Qin, Xiaoxing Wang, and Junchi Yan. Revisiting fairness in multitask learning: A performance-driven approach for variance reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20492–20501, 2025.

[65] 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*, pages 10067–10076, 2025.

[66] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2017.

[67] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

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

[69] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4822–4829, 2019.

[70] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[71] 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*, pages 20083–20093, 2023.

[72] Jiayi Shen, Xiantong Zhen, Marcel Worring, and Ling Shao. Variational multi-task learning with gumbel-softmax priors. In *Advances in Neural Information Processing Systems*, pages 21031–21042, 2021.

[73] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, pages 746–760, 2012.

[74] Max Simchowitz and Kevin G Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. In *Advances in Neural Information Processing Systems*, pages 1151–1160, 2019.

[75] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[76] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *Proceedings of the International Conference on Machine Learning*, pages 9120–9132, 2020.

[77] Guolei Sun, Thomas Probst, Danda Pani Paudel, Nikola Popović, Menelaos Kanakis, Jagruti Patel, Dengxin Dai, and Luc Van Gool. Task switching network for multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8291–8300, 2021.

[78] Sebastian Thrun and Joseph O'sullivan. Discovering structure in multiple learning tasks: The tc algorithm. In *Proceedings of the International Conference on Machine Learning*, pages 489 – 497, 1996.

[79] Berk Ustun and Cynthia Rudin. Learning optimized risk scores. *Journal of Machine Learning Research*, 20(150):1–75, 2019.

[80] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3614–3633, 2021.

[81] Hang Wang, Sen Lin, and Junshan Zhang. Warm-start actor-critic: From approximation error to sub-optimality gap. In *Proceedings of the International Conference on Machine Learning*, pages 35989–36019, 2023.

[82] Xiao Wang, Yuying Han, Dazi Li, Fei Zhang, and Min Tang. From gradient volume to shapley fairness: Towards fair multi-task learning. In *The International Conference on Learning Representations*, 2026.

[83] Yifei Wang, Tolga Ergen, and Mert Pilanci. Parallel deep neural networks have zero duality gap. In *The International Conference on Learning Representations*, 2023.

[84] Zedong Wang, Siyuan Li, and Dan Xu. Rep-mtl: Unleashing the power of representation-level task saliency for multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3413–3423, 2025.

[85] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 2011.

[86] Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. In *Advances in Neural Information Processing Systems*, pages 4509–4533, 2023.

[87] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

[88] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, pages 5824–5836, 2020.

[89] Amir R Zamir, Alexander Sax, William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[90] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

[91] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Proceedings of the European Conference on Computer Vision*, pages 94–108, 2014.

[92] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4352–4360. IEEE, 2017.

Appendix

This appendix is organized as follows:

- Section A provides the proof of the closed-form solution under the isotropic approximation.
- Section B proves finite termination and optimality of the inner active-set loop in Algorithm 1.
- Section C states the convergence guarantee for the outer training iteration.
- Section D derives the two-task angle–rate relationship under the isotropic approximation.
- Section E motivates the gradient-norm curvature proxy under explicit local assumptions.
- Section F presents implementation details, additional experimental results and ablations, and active-set analysis.
- Section G discusses the social impact of the proposed method.

A Proof of the Closed-Form Solution

We derive the solution in Section 3.5 for fixed gradients with $\|g_i\| > 0$ and analytical step size $\alpha > 0$. We first solve the unregularized isotropic surrogate, then introduce Gram regularization.

Proof. With $H_i \approx \gamma_i I$, $\gamma_i = \|g_i\|$, and $g_i = \nabla l_i(\theta_t)$, we have $\beta_i = \|g_i\|/2$ and $\hat{g}_i = g_i/(2\beta_i) = g_i/\|g_i\|$. Equation (12) becomes

$$\widehat{r}_i(d_t) = \frac{\alpha g_i^\intercal d_t - \alpha^2 \beta_i \|d_t\|^2}{\beta_i} = 2\alpha \hat{g}_i^\intercal d_t - \alpha^2 \|d_t\|^2, \tag{20}$$

The unregularized max-min surrogate is

$$\min_{d_t,\tau} -\tau \qquad \text{subject to} \qquad (\tau - \widehat{r}_i(d_t)) \le 0, \quad i \in [K]. \tag{21}$$

This problem is convex and satisfies Slater's condition: $d_t = 0$ and $\tau = -1$ are strictly feasible. Thus, strong duality holds and the KKT conditions are necessary and sufficient. Let $A = \{i : \lambda_i^* > 0\}$ be the nonempty support of an optimal multiplier vector. Its reduced Lagrangian is

$$\mathcal{L}_A(d_t, \tau, \lambda_A) = -\tau + \sum_{i \in A} \lambda_i \big(\tau - \widehat{r}_i(d_t)\big). \tag{22}$$

Substituting Eq. (20) gives

$$\mathcal{L}_A(d_t, \tau, \lambda_A) = \Big(\sum_{i \in A} \lambda_i - 1\Big)\tau - 2\alpha \Big(\sum_{i \in A} \lambda_i \hat{g}_i\Big)^\intercal d_t + \alpha^2 \Big(\sum_{i \in A} \lambda_i\Big)\|d_t\|^2. \tag{23}$$

Stationarity in τ gives

$$\sum_{i \in A} \lambda_i^* = 1. \tag{24}$$

and stationarity in d_t gives

$$d_t^* = \frac{1}{\alpha} \sum_{i \in A} \lambda_i^* \hat{g}_i. \tag{25}$$

Write $c_A^* \coloneqq \sum_{i \in A} \lambda_i^* \hat{g}_i = \alpha d_t^*$. For every task $i \in [K]$,

$$\widehat{r}_i(d_t^*) = 2\hat{g}_i^\intercal c_A^* - \|c_A^*\|^2. \tag{26}$$

Let G_A be the submatrix of G indexed by A. Complementary slackness gives $\widehat{r}_i(d_t^*) = \tau_A^*$ for $i \in A$, where τ_A^* is the unregularized worst-case surrogate rate. Taking the weighted sum in Eq. (26) yields

$$\tau_A^* = \sum_{i \in A} \lambda_i^* \widehat{r}_i(d_t^*) = \|c_A^*\|^2 = \lambda_A^{*\intercal} G_A \lambda_A^*. \tag{27}$$

Substituting back into the active constraints gives $G_A\lambda_A^* = \tau_A^*\mathbf{1}_A$. If G_A is invertible, normalization therefore yields

$$s_A^{(0)} \coloneqq \mathbf{1}_A^\intercal G_A^{-1}\mathbf{1}_A, \qquad \tau_A^* = \frac{1}{s_A^{(0)}}, \qquad \lambda_A^* = \frac{G_A^{-1}\mathbf{1}_A}{s_A^{(0)}}. \tag{28}$$

For a candidate set, these formulas give a global solution only if $\lambda_A^* \geq 0$ and $\widehat{r}_j(d_t^*) \geq \tau_A^*$ for all $j \notin A$. A zero multiplier does not require a strict rate inequality. If G_A is singular, the inverse formula does not apply.

To obtain the regularized solve, note that minimizing the full Lagrangian over (d_t, τ) gives $-\lambda^\intercal G\lambda$ when $\lambda \geq 0$ and $\mathbf{1}^\intercal\lambda = 1$. Thus, the unregularized dual is equivalent to minimizing $\lambda^\intercal G\lambda$ on the simplex. Adding $\rho\|\lambda\|^2$, with $\rho > 0$, gives $\tilde{G} = G + \rho I \succ 0$. On a nonempty candidate set A, equality-constrained stationarity and normalization give the main-text formulas

$$s_A = \mathbf{1}_A^\intercal\tilde{G}_A^{-1}\mathbf{1}_A, \qquad \lambda_A^* = \frac{\tilde{G}_A^{-1}\mathbf{1}_A}{s_A}, \qquad \tau_A^* = \lambda_A^{*\intercal}\tilde{G}_A\lambda_A^* = \frac{1}{s_A}. \tag{29}$$

Here, τ_A^* denotes the regularized objective value, not generally the achieved worst-case surrogate rate. With $\lambda_j^* = 0$ for $j \notin A$, the candidate is globally optimal if $\lambda_A^* \geq 0$ and $(\tilde{G}\lambda^*)_j \geq \tau_A^*$ for all $j \notin A$, as proved in Appendix B. The analytical direction is then recovered from Eq. (18). □

B Termination and Optimality of the Active-Set Loop

For fixed task gradients, the active-set loop of Algorithm 1 solves the regularized quadratic program

$$\min_{\lambda\in\mathbb{R}^K} \quad \phi(\lambda) \coloneqq \lambda^\intercal\tilde{G}\lambda \tag{30}$$

$$\text{subject to} \quad \lambda \geq 0, \qquad \mathbf{1}^\intercal\lambda = 1, \tag{31}$$

where $\tilde{G} = G + \rho I$ and $\rho > 0$. Since $\tilde{G} \succ 0$, the objective is strictly convex on the simplex.

Proposition 1. *Assume $\|g_i\| > 0$ for all $i \in [K]$, $\rho > 0$, and exact arithmetic. With the no-repeat rule and lexicographic fallback in Algorithm 1, the active-set loop finds the unique minimizer of Eq.* (31) *after at most $2^K - 1$ candidate solves. These multipliers uniquely determine the returned vector $\sum_i \lambda_i^*\hat{g}_i$ and the analytical direction d_t^* in Eq.* (18)*.*

Proof. Fix a nonempty candidate set $A \subseteq [K]$ and let $\Delta_A \coloneqq \{\lambda \geq 0 : \mathbf{1}^\intercal\lambda = 1,\ \lambda_j = 0 \text{ for } j \notin A\}$ be its simplex face. With $s_A = \mathbf{1}_A^\intercal\tilde{G}_A^{-1}\mathbf{1}_A$, the unique minimizer on the affine hull of Δ_A is

$$\lambda_A^* = \frac{\tilde{G}_A^{-1}\mathbf{1}_A}{s_A}, \qquad \tau_A^* = \phi(\lambda^*) = \frac{1}{s_A}, \tag{32}$$

with $\lambda_j^* = 0$ for $j \notin A$. If λ_A^* is componentwise nonnegative, it is also the minimizer over the face Δ_A. The first-order condition on the affine hull gives

$$(\tilde{G}\lambda^*)_i = \tau_A^*, \qquad i \in A. \tag{33}$$

A negative component makes the candidate infeasible. The proposed pivot removes one such index; the resulting set is nonempty because $\mathbf{1}_A^\intercal\lambda_A^* = 1$.

If $\lambda_A^* \geq 0$ but an inactive index $j \notin A$ violates the optimality condition, $(\tilde{G}\lambda^*)_j < \tau_A^*$, consider the feasible perturbation

$$\lambda(\varepsilon) = (1-\varepsilon)\lambda^* + \varepsilon e_j, \qquad 0 \leq \varepsilon \leq 1, \tag{34}$$

where e_j is the jth basis vector. Using $\mathbf{1}^\intercal\lambda^* = 1$ and (33),

$$\frac{d}{d\varepsilon}\phi(\lambda(\varepsilon))\bigg|_{\varepsilon=0} = 2\big((\tilde{G}\lambda^*)_j - \lambda^{*\intercal}\tilde{G}\lambda^*\big) = 2\big((\tilde{G}\lambda^*)_j - \tau_A^*\big) < 0. \tag{35}$$

Thus, a sufficiently small step toward e_j decreases the objective, justifying the proposed addition of j. This does not imply that j must belong to the final support.

If both tests pass, the KKT conditions hold: with equality multiplier $\nu = -2\tau_A^*$ and nonnegativity multipliers $\mu_i = 2[(\tilde{G}\lambda^*)_i - \tau_A^*]$, we have $2\tilde{G}\lambda^* + \nu\mathbf{1} - \mu = 0$, $\mu \geq 0$, and $\mu_i\lambda_i^* = 0$. Strict convexity makes this candidate the unique global minimizer.

For termination, the compact simplex and strict convexity ensure a unique minimizer $\bar{\lambda}$. Its nonempty support $\bar{A} = \{i : \bar{\lambda}_i > 0\}$ satisfies the KKT conditions. Stationarity on $\bar{A}$ and normalization imply that $\bar{\lambda}$ is exactly the equality-constrained candidate for $\bar{A}$, so this set passes both tests. Until termination, $\bar{A}$ must therefore remain unvisited. The no-repeat rule and lexicographic fallback always select an unvisited nonempty set, so the loop terminates after at most $2^K - 1$ candidate solves. The unique multipliers determine both the returned vector and d_t^* through Eq. (18). □

C Convergence of the Training Iteration

We analyze convergence of the training iterates under full gradients, an exact unregularized inner solve ($\rho = 0$), and a diminishing outer step size. The result does not cover fixed positive Gram regularization, stochastic gradients, momentum, or adaptive preconditioning.

To make the iteration dependence explicit, write $g_{i,t} = \nabla l_i(\theta_t)$, $\beta_{i,t} = \|g_{i,t}\|/2$, and $\hat{g}_{i,t} = g_{i,t}/(2\beta_{i,t})$. Using the aggregated gradient returned by the unregularized inner solve, we study the update with outer step size $\eta_t > 0$,

$$\theta_{t+1} = \theta_t - \eta_t \sum_{i=1}^{K} \lambda_{i,t}^* \hat{g}_{i,t}. \tag{36}$$

The unregularized inner solution is defined by the convex-hull problem below, without requiring an invertible Gram matrix.

Proposition 2. *Assume each loss l_i is continuously differentiable and bounded below, with L-Lipschitz gradient on a neighborhood containing the closure of the iterates and all update segments. Suppose $\beta_{i,t} \geq \underline{\beta} > 0$ for every i, t, and the update in Eq.* (36) *uses*

$$\sum_{i=1}^{K} \lambda_{i,t}^* \hat{g}_{i,t} = \arg\min_{z \in C_t} \|z\|^2, \qquad C_t \coloneqq \operatorname{conv}\{\hat{g}_{i,t} : i \in [K]\}, \tag{37}$$

with step sizes satisfying

$$\eta_t > 0, \qquad \sum_{t=0}^{\infty} \eta_t = \infty, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty. \tag{38}$$

Then $\|\sum_i \lambda_{i,t}^* \hat{g}_{i,t}\| \to 0$. Every accumulation point $\bar{\theta}$ is Pareto stationary: for $\beta_i(\theta) \coloneqq \|\nabla l_i(\theta)\|/2$, there exists $\bar{\lambda} \geq 0$ with $\mathbf{1}^\intercal\bar{\lambda} = 1$ such that

$$\sum_{i=1}^{K} \bar{\lambda}_i \frac{\nabla l_i(\bar{\theta})}{2\beta_i(\bar{\theta})} = 0. \tag{39}$$

This does not assert that an accumulation point exists or that the full parameter sequence converges.

Proof. Choose convex weights representing the unique projection onto C_t, so that $\lambda_{i,t}^* \geq 0$ and $\sum_i \lambda_{i,t}^* = 1$. Define its squared norm by

$$h_t \coloneqq \left\| \sum_{i=1}^{K} \lambda_{i,t}^* \hat{g}_{i,t} \right\|^2. \tag{40}$$

Since $\|\hat{g}_{i,t}\| = 1$, we have $0 \leq h_t \leq 1$. Projection optimality gives, for every $z \in C_t$,

$$\left\langle \sum_{i=1}^{K} \lambda_{i,t}^* \hat{g}_{i,t}, z \right\rangle \geq h_t. \tag{41}$$

Taking $z = \hat{g}_{i,t}$ yields

$$\nabla l_i(\theta_t)^\intercal \sum_{j=1}^{K} \lambda_{j,t}^* \hat{g}_{j,t} \geq 2\underline{\beta} h_t. \tag{42}$$

Let $F(\theta) = \sum_{i=1}^{K} l_i(\theta)$. By L-smoothness of each task loss and the update in Eq. (36),

$$F(\theta_{t+1}) \leq F(\theta_t) - \eta_t \sum_{i=1}^{K} \nabla l_i(\theta_t)^\intercal \sum_{j=1}^{K} \lambda_{j,t}^* \hat{g}_{j,t} + \frac{KL}{2} \eta_t^2 h_t \tag{43}$$

$$\leq F(\theta_t) - \left(2K\underline{\beta}\eta_t - \frac{KL}{2}\eta_t^2 \right) h_t. \tag{44}$$

Since $\eta_t \to 0$, for all sufficiently large t the coefficient is at least $K\underline{\beta}\eta_t$. Because F is bounded below, summing the descent inequality gives

$$\sum_{t=0}^{\infty} \eta_t h_t < \infty. \tag{45}$$

To prove $h_t \to 0$, gradient Lipschitz continuity and $\|g_{i,t}\| \geq 2\underline{\beta}$ give

$$\|\hat{g}_{i,t+1} - \hat{g}_{i,t}\| \leq \frac{\|g_{i,t+1} - g_{i,t}\|}{\underline{\beta}} \leq \frac{L}{\underline{\beta}} \|\theta_{t+1} - \theta_t\|. \tag{46}$$

Using the same convex weights in C_t and C_{t+1} bounds the change in their distances to the origin by the largest change in a vertex. Since both distances are at most one,

$$|h_{t+1} - h_t| \leq \frac{2L}{\underline{\beta}} \|\theta_{t+1} - \theta_t\| \leq C\eta_t, \qquad C \coloneqq 1 + \frac{2L}{\underline{\beta}}. \tag{47}$$

Equation (45) and $\sum_t \eta_t = \infty$ imply $\liminf_t h_t = 0$. If h_t did not converge to zero, some $\varepsilon > 0$ would admit infinitely many disjoint intervals $[a, b)$ with $h_a \geq 2\varepsilon$, $h_b < \varepsilon$, and $h_t \geq \varepsilon$ for $a \leq t < b$. Each interval would satisfy $\sum_{t=a}^{b-1} \eta_t \geq \varepsilon/C$ and hence $\sum_{t=a}^{b-1} \eta_t h_t \geq \varepsilon^2/C$, contradicting Eq. (45). Therefore $h_t \to 0$.

Let $\theta_{t_s} \to \bar{\theta}$. By continuity, $\beta_i(\bar{\theta}) \geq \underline{\beta}$, so all normalized gradients are continuous at $\bar{\theta}$. Compactness of the simplex gives a further subsequence with $\lambda_{t_s}^* \to \bar{\lambda} \geq 0$ and $\mathbf{1}^\intercal \bar{\lambda} = 1$. Since $h_{t_s} \to 0$, passing to the limit in $\sum_i \lambda_{i,t_s}^* \hat{g}_{i,t_s}$ proves Eq. (39). □

The uniform lower bound on $\beta_{i,t}$ excludes trajectories along which any task gradient vanishes. The theorem is therefore conditional and does not cover all training trajectories. Its conclusion rules out a common strict first-order descent direction at an accumulation point, but does not guarantee a local minimum or global optimum for nonconvex losses.

Role of β_i in the limit point. Positive normalization changes the search direction but preserves the Pareto-stationary set. Specifically, Eq. (39) is equivalent to

$$0 \in \operatorname{conv} \left\{ \nabla l_i(\bar{\theta}) : i \in [K] \right\}, \tag{48}$$

provided all $\beta_i(\bar{\theta})$ are finite and positive. Indeed, the weights

$$\nu_i = \frac{\bar{\lambda}_i / (2\beta_i(\bar{\theta}))}{\sum_j \bar{\lambda}_j / (2\beta_j(\bar{\theta}))} \tag{49}$$

satisfy $\nu \geq 0$, $\mathbf{1}^\intercal \nu = 1$, and $\sum_i \nu_i \nabla l_i(\bar{\theta}) = 0$; the converse uses $\bar{\lambda}_i \propto 2\beta_i(\bar{\theta})\nu_i$. For convex losses, this makes $\bar{\theta}$ a minimizer of a nonnegative weighted sum of losses and hence a weak Pareto optimum. With $\beta_i = \|\nabla l_i\|/2$, the inner solve uses unit task gradients: normalization can change the trajectory and the stationary point reached, but not the stationary set where all task gradients are nonzero.

D Two-Task Angle–Rate Relationship Under the Isotropic Approximation

For two tasks, the isotropic surrogate rate has an explicit dependence on the angle between task gradients. Fix nonzero gradients $g_1, g_2 \in \mathbb{R}^m$ and let $\vartheta \in [0, \pi]$ be their angle. With $\gamma_i = \|g_i\|$ and $\beta_i = \|g_i\|/2$, the normalized gradients are

$$\hat{g}_i \coloneqq \frac{g_i}{2\beta_i} = \frac{g_i}{\|g_i\|}, \qquad i \in \{1, 2\}. \tag{50}$$

Hence each $\hat{g}_i$ has unit norm, and their Gram matrix is

$$G = \begin{bmatrix} \hat{g}_1^\intercal \hat{g}_1 & \hat{g}_1^\intercal \hat{g}_2 \\ \hat{g}_2^\intercal \hat{g}_1 & \hat{g}_2^\intercal \hat{g}_2 \end{bmatrix} = \begin{bmatrix} 1 & \cos\vartheta \\ \cos\vartheta & 1 \end{bmatrix}, \tag{51}$$

For the unregularized dual in Appendix A, write $\lambda_2 = 1 - \lambda_1$ with $0 \le \lambda_1 \le 1$. Then

$$\lambda^\intercal G \lambda = \frac{1 + \cos\vartheta}{2} + 2(1 - \cos\vartheta)\left(\lambda_1 - \frac{1}{2}\right)^2. \tag{52}$$

Thus, for $0 < \vartheta \le \pi$, the unique optimal weights are

$$\lambda_1^* = \lambda_2^* = \frac{1}{2}. \tag{53}$$

At $\vartheta = 0$, every feasible weight pair is optimal and gives the same aggregated gradient; we choose equal weights. This argument also covers the singular Gram matrices at both endpoints. Equation (18) gives

$$d_t^* = \frac{\hat{g}_1 + \hat{g}_2}{2\alpha}. \tag{54}$$

For either task $i \in \{1, 2\}$,

$$\hat{g}_i^\intercal(\hat{g}_1 + \hat{g}_2) = 1 + \cos\vartheta, \qquad \|\hat{g}_1 + \hat{g}_2\|^2 = 2(1 + \cos\vartheta). \tag{55}$$

Substituting into Eq. (20) yields the optimal surrogate rates

$$\widehat{r}_1(d_t^*) = \widehat{r}_2(d_t^*) = \frac{1 + \cos\vartheta}{2}. \tag{56}$$

The surrogate rate is therefore affine in $\cos\vartheta$: aligned, orthogonal, and opposite gradients give rates 1, $1/2$, and 0, respectively. This identity concerns the quadratic model, rather than the exact loss-based rate r_i in Eq. (5).

For $\vartheta < \pi$, substituting the surrogate rate into the gap relation gives, for either task,

$$\frac{1}{\widehat{r}_i(d_t^*)} - 1 = \frac{1 - \cos\vartheta}{1 + \cos\vartheta}. \tag{57}$$

This surrogate gap increases with ϑ and diverges as $\vartheta \to \pi$. At $\vartheta = \pi$, it is undefined because the surrogate improvement is zero.

Effect of diagonal regularization. Section 3.5 uses $\tilde{G} = G + \rho I$ with $\rho > 0$. For two tasks,

$$\tilde{G} = \begin{bmatrix} 1 + \rho & \cos\vartheta \\ \cos\vartheta & 1 + \rho \end{bmatrix}, \tag{58}$$

The regularized objective is

$$\lambda^\intercal \tilde{G} \lambda = \frac{1 + \rho + \cos\vartheta}{2} + 2(1 + \rho - \cos\vartheta)\left(\lambda_1 - \frac{1}{2}\right)^2. \tag{59}$$

Its unique minimizer is $\lambda_1^* = \lambda_2^* = 1/2$ for every $\vartheta \in [0, \pi]$. Hence, for $A = \{1, 2\}$, the main-text objective value is

$$\tau_A^* = \lambda^{*\intercal} \tilde{G} \lambda^* = \frac{1 + \rho + \cos\vartheta}{2}. \tag{60}$$

Regularization leaves the aggregated gradient and the surrogate rates in Eq. (56) unchanged, while increasing the objective value by $\rho/2$. Thus, τ_A^* can exceed 1; it is a regularized objective value, not a normalized improvement rate.

E Local Justification of the Isotropic Curvature Scale

Following the local Hessian-approximation perspective of [55], assume l_i is twice continuously differentiable near a stationary reference $\theta_i^\star$ and the recent iterates. Fix iteration t, with $g_i = \nabla l_i(\theta_t)$ and $H_i = \nabla^2 l_i(\theta_t)$ as in the main text. For an index k in a recent window $\mathcal{W}_t$, define $g_{i,k} = \nabla l_i(\theta_k)$ and $\delta_{i,k} = \theta_k - \theta_i^\star$. If the Hessian remains close to H_i along the segments joining $\theta_i^\star$ to these iterates, Taylor expansion gives $g_{i,k} \approx H_i \delta_{i,k}$.

Let $\mathbb{E}_t$ denote the uniform average over $k \in \mathcal{W}_t$. Assume the displacement second moment is approximately isotropic,

$$\mathbb{E}_t[\delta_{i,k}\delta_{i,k}^\intercal] \approx \sigma_i^2 I, \tag{61}$$

where $\sigma_i > 0$ is a task-specific displacement scale for this window. This is an uncentered second-moment assumption, rather than a covariance assumption. Treating H_i as constant gives

$$\mathbb{E}_t[g_{i,k}g_{i,k}^\intercal] \approx H_i \mathbb{E}_t[\delta_{i,k}\delta_{i,k}^\intercal] H_i^\intercal \approx \sigma_i^2 H_i H_i^\intercal, \tag{62}$$

$$\mathbb{E}_t\|g_{i,k}\|^2 \approx \sigma_i^2 \operatorname{tr}(H_i H_i^\intercal) = \sigma_i^2 \|H_i\|_F^2. \tag{63}$$

Thus, these moments reflect curvature magnitude, not its sign. Under the additional positive isotropic approximation $H_i \approx \gamma_i I$ in the m-dimensional parameter space,

$$\mathbb{E}_t\|g_{i,k}\|^2 \approx \sigma_i^2 m \gamma_i^2, \qquad \gamma_i \approx \frac{\sqrt{\mathbb{E}_t\|g_{i,k}\|^2}}{\sigma_i \sqrt{m}}, \qquad \gamma_i > 0. \tag{64}$$

Comparable displacement scales across tasks and the approximation $\sqrt{\mathbb{E}_t\|g_{i,k}\|^2} \approx \|g_i\|$ motivate the proportional proxy $\gamma_i \propto \|g_i\|$. Choosing the common proportionality constant to be one gives $\gamma_i = \|g_i\|$ and the unit-length isolated benchmark in Section 3.5; this choice is a modeling convention, not a consequence of the moment identity.

The argument motivates a tractable surrogate under local assumptions. Anisotropic displacements, unequal task-specific displacement scales, varying or indefinite curvature, and stochastic gradient noise limit its applicability. Empirical comparisons are provided in Appendices F.4 and F.5.

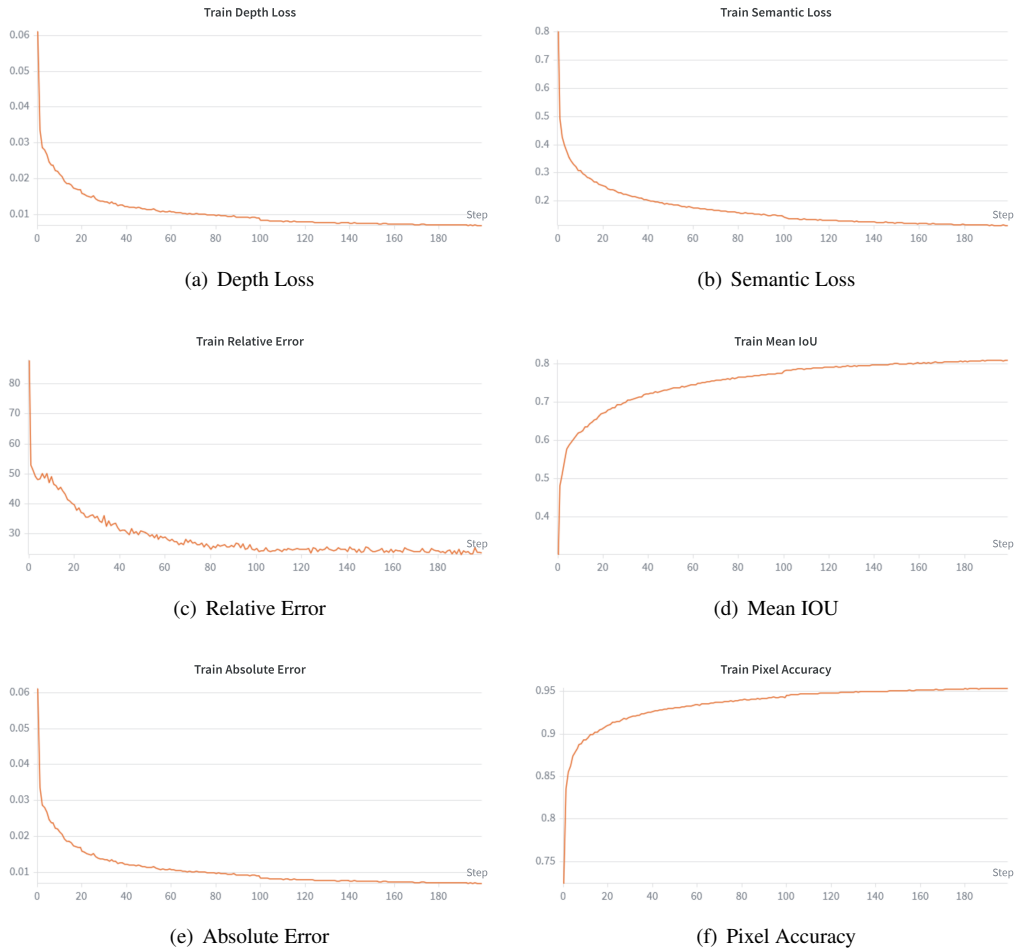
F Experiments and Results

F.1 Implementation Details

CelebA. To evaluate performance in a many-task setting, we use CelebA [50], which contains more than 200K celebrity face images. Since each image is annotated with 40 attributes, such as smiling, wavy hair, and mustache, we formulate this benchmark as an image-level 40-task MTL classification problem in which each task corresponds to one attribute. Our implementation follows [46, 4]. Specifically, the network consists of a 9-layer convolutional neural network (CNN) backbone and a separate linear layer for each task. Training is performed for 15 epochs with Adam optimizer. We set learning rate and batch size to be 3e-4 and 256 respectively.

QM9. For graph-structured data, we consider QM9 [7], a widely used benchmark comprising more than 130k organic molecules represented as graphs with node and edge features. The task is to predict 11 properties measured on different scales, making this dataset suitable for assessing whether MTL methods can handle heterogeneous tasks. Consistent with [46, 4], we use the Pytorch Geometric example implementation [25]. The data split uses 110k molecules for training, 10k for validation, and the remaining 10k for testing. We train for 300 epochs with batch size 120, start from a learning rate of 1e-3, and reduce it with a scheduler whenever validation progress plateaus.

NYU-V2 and CityScapes. To study more complex dense prediction settings, we use NYU-v2 [73] and Cityscapes [19]. NYU-v2 contains 1449 densely annotated indoor-scene images collected from video sequences and includes three tasks: 13-class semantic segmentation, depth estimation, and surface normal prediction, corresponding to one pixel-level classification task and two pixel-level regression tasks. Cityscapes contains 5000 street-scene images and involves two tasks, 7-class semantic segmentation and depth estimation. Following [46, 4], we adopt the MTAN backbone [49], which equips SegNet [3] with task-specific attention modules. We train the model for 200 epochs, using batch size 2 on NYU-v2 and batch size 8 on Cityscapes. The learning rate remains 1e-4 for the first 100 epochs and is then halved for the rest of training.



(a) Depth Loss (b) Semantic Loss

(c) Relative Error (d) Mean IOU

(e) Absolute Error (f) Pixel Accuracy

Figure 6: Evolution of training losses and evaluation metrics for our method on Cityscapes: **(a)** depth loss, **(b)** semantic segmentation loss, **(c)** relative error, **(d)** mean intersection over union (mIoU), **(e)** absolute error, and **(f)** pixel accuracy.

UTKFace. We compare our method with BayesAgg-MTL [1] on the UTKFace dataset [92], which contains over 20,000 face images annotated with age, gender, and ethnicity. We formulate age prediction as a regression task, with ages ranging from 0 to 116 years; gender prediction as binary classification (male or female); and ethnicity prediction as five-class classification. Following the experimental setup of BayesAgg-MTL, we split the dataset into training, validation, and test sets comprising 70%, 10%, and 20% of the images, respectively, and use ResNet-18 as the shared backbone.

F.2 Training Loss Curves

Figure 6 shows the evolution of the training losses and evaluation metrics on the Cityscapes dataset, including depth loss, semantic segmentation loss, relative error, mean intersection over union (mIoU), absolute error, and pixel accuracy. The training loss curves exhibit smooth trajectories.

F.3 Comparison with Normalized-Gradient Variants

To examine the effect of gradient normalization alone, we evaluate normalized variants of MGDA [70], IMTL-G [48], and NashMTL [58] on NYU-v2, QM9, and CelebA. We use MGDA's default gradient normalization scheme, while applying the same normalization as in our method to IMTL-G and NashMTL. Table 5 also reports results for the original IMTL-G and NashMTL methods for reference.

Normalization increases the mean $\boldsymbol{\Delta m}$ for IMTL-G on all three datasets and for NashMTL on NYU-v2 and QM9, but decreases it for NashMTL on CelebA. Our method achieves the lowest reported mean $\boldsymbol{\Delta m}$ on all three datasets. These results indicate that gradient normalization alone does not consistently improve the performance of existing methods.

Table 5: Ablations on normalized-gradient variants.

Method	**Normalize**	NYU-v2 $\boldsymbol{\Delta m} \downarrow$	QM9 $\boldsymbol{\Delta m} \downarrow$	CelebA $\boldsymbol{\Delta m} \downarrow$
MGDA	✓	1.38	120.5	10.93
IMTL-G		-0.76	77.2	4.67
IMTL-G	✓	2.63	1674.4	37.31
(error bar)		($\pm$2.31)	($\pm$166.31)	($\pm$7.55)
NashMTL		-4.04	62.0	4.97
NashMTL	✓	2.86	155.8	2.24
(error bar)		($\pm$0.75)	($\pm$14.66)	($\pm$1.55)
Ours	✓	$\mathbf{-5.49}$	$\mathbf{54.3}$	$\mathbf{-0.34}$
(error bar)		($\pm$0.74)	($\pm$1.50)	($\pm$0.52)

F.4 Comparison with a Diagonal Hessian Surrogate

We compare our isotropic approximation with the more expressive damped diagonal surrogate $D_i = \mathrm{diag}(g_i g_i^\intercal) + \varepsilon I$, where $\varepsilon > 0$ ensures positive definiteness. Because the aggregated curvature depends on both the coordinates and the task multipliers, the original Gram-matrix closed-form solution no longer applies; we instead optimize the multipliers by projected gradient descent over the simplex. As shown in Table 6, the diagonal variant yields a substantially higher mean $\boldsymbol{\Delta m}$ on NYU-v2, indicating worse performance in this experiment. One possible explanation is that raw squared-gradient diagonals are more sensitive to stochastic coordinate-wise variation and near-zero entries, while also requiring additional inner-loop optimization. However, this experiment does not isolate the contributions of these factors. These results suggest that a more expressive curvature surrogate does not necessarily improve multi-task balancing and support our use of the simpler isotropic approximation, while motivating further work on stable and efficient anisotropic surrogates.

Table 6: Hessian-surrogate comparison on NYU-v2.

Hessian Approximation	$\boldsymbol{\Delta m} \downarrow$
$\mathrm{diag}(g_i g_i^\intercal) + \varepsilon I$	-0.76 ± 0.28
Isotropic	$\mathbf{-5.49} \pm 0.74$

F.5 Additional Ablations on NYU-v2 and QM9

We extend the Cityscapes ablations to NYU-v2 (3 tasks) and QM9 (11 tasks) to assess the roles of the regularization strength ρ, isotropic curvature scale γ_i, and task-specific normalization factor β_i beyond the two-task setting. As shown in Table 7, the default configuration achieves the lowest reported mean $\boldsymbol{\Delta m}$ on both datasets across all three ablations. Replacing $\gamma_i = \|g_i\|$ with 1 or $\|g_i\|^2$, or setting $\beta_i = 1$, degrades performance, particularly on QM9.

The term ρI is a deterministic diagonal regularizer that makes the mini-batch Gram matrix positive definite and reduces sensitivity to ill-conditioning. For nonzero task gradients normalized to unit norm, $G_{ii} = 1$, so $\rho / G_{ii} = \rho$. Among the tested values, $\rho = 1.5$ yields a lower mean $\boldsymbol{\Delta m}$ than either weaker ($\rho = 0.15$) or stronger ($\rho = 15$) regularization on both datasets, with a more pronounced advantage on QM9.

F.6 Comparison with BayesAgg-MTL on UTKFace

We further compare our method with BayesAgg-MTL [1] on UTKFace under matched experimental settings for age, gender, and ethnicity prediction. As shown in Table 8, the two methods tie on age

Table 7: Ablations on NYU-v2 and QM9.

Component	**Setting**	NYU-v2 $\boldsymbol{\Delta m} \downarrow$	QM9 $\boldsymbol{\Delta m} \downarrow$
Regularization	$\rho = 0.15$	-5.32 ± 0.50	73.9 ± 1.68
	$\rho = 1.5$	$\mathbf{-5.49} \pm 0.74$	$\mathbf{54.3} \pm 1.50$
	$\rho = 15$	-5.02 ± 1.21	59.4 ± 6.36
Hessian approximation	$\gamma_i = 1$	-4.62 ± 0.85	94.3 ± 4.11
	$\gamma_i = \|g_i\|$	$\mathbf{-5.49} \pm 0.74$	$\mathbf{54.3} \pm 1.50$
	$\gamma_i = \|g_i\|^2$	-0.26 ± 2.30	249.5 ± 6.53
Task normalization	$\beta_i = 1$	-2.63 ± 0.98	162.2 ± 7.27
	Ours	$\mathbf{-5.49} \pm 0.74$	$\mathbf{54.3} \pm 1.50$

prediction at the reported precision, while our method achieves higher scores on gender and ethnicity prediction. Our method also achieves a slightly lower $\boldsymbol{\Delta m}$ and a lower reported mean rank.

Table 8: Comparison with BayesAgg-MTL on UTKFace.

Method	Age $\downarrow$	Gender $\uparrow$	Ethnicity $\uparrow$	$\boldsymbol{\Delta m} \downarrow$	**MR**$\downarrow$
BayesAgg-MTL	$\mathbf{1.35}$	93.01	84.25	-2.23	3.33
Ours	$\mathbf{1.35}$	$\mathbf{93.39}$	$\mathbf{84.70}$	$\mathbf{-2.30}$	$\mathbf{1.00}$
(error bar)	($\pm$0.01)	($\pm$0.23)	($\pm$0.21)	($\pm$0.09)	

F.7 Active-Set Analysis on CelebA

To examine active-set behavior in a many-task setting, we analyze its size and task-level activation rates on CelebA when $\rho = 0.15$. We record the active-set size at each iteration and compute its mean and standard deviation within each epoch. For each task, we also compute its activation rate, defined as the fraction of iterations within an epoch in which the task belongs to the active set. Table 9 reports these size statistics together with the minimum, mean, and maximum activation rates across the 40 tasks.

Table 9: Active-set statistics on CelebA across training epochs. Activation rates are summarized across the 40 tasks.

Epoch	**Active Tasks**	**Activation Rate**		
		Minimum	Mean	Maximum
1	35.39 ± 3.28	0.57	0.88	0.98
2	36.46 ± 2.75	0.73	0.91	0.97
3	36.29 ± 2.98	0.77	0.91	0.96
4	36.83 ± 2.52	0.82	0.92	0.97
5	36.78 ± 2.72	0.83	0.92	0.97
6	36.53 ± 3.04	0.84	0.91	0.96
7	37.07 ± 2.67	0.87	0.93	0.97
8	37.18 ± 2.36	0.86	0.93	0.97
9	36.84 ± 2.82	0.86	0.92	0.97
10	37.34 ± 2.46	0.88	0.93	0.98
11	37.28 ± 2.48	0.88	0.93	0.97
12	37.37 ± 2.33	0.86	0.93	0.97
13	37.83 ± 2.22	0.90	0.94	0.98
14	37.74 ± 2.21	0.89	0.94	0.98
15	37.38 ± 2.50	0.88	0.93	0.97

The active set contains most, but not always all, of the 40 tasks. Its average size increases from 35.39 in the first epoch to approximately 37 in later epochs. The minimum task-level activation rate also increases from 0.57 to approximately 0.9, while the maximum remains between 0.96 and 0.98. This narrowing of the range of activation rates indicates that task participation becomes more balanced as training progresses.

G Social Impact

This work is primarily foundational research on optimization for multi-task learning rather than a deployment-specific system. Its main potential positive impact is to improve how shared models balance progress across tasks, which may increase reliability and data efficiency in applications such as scene understanding, scientific prediction, and assistive systems. Because the method can improve shared optimization without requiring separate models for each task, it may also reduce computational cost and energy consumption in settings where multi-task learning is appropriate.

Misuse is also possible. For example, the method could be incorporated into systems that jointly perform detection, tracking, identification, and attribute inference, making privacy-invasive or surveillance-oriented pipelines easier to train and scale. Appropriate mitigation strategies include monitoring for misuse when releasing code, documenting intended-use limitations, and using human oversight when model outputs affect consequential decisions. Because this paper contributes an optimization method rather than a trained high-risk model, we believe open scientific discussion is appropriate, but downstream deployments should still be subject to application-specific risk assessment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The method section gives derivation, and the experiment section demonstrates the effectiveness of the method.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The main text discusses curvature approximation, many-task costs, the theory–implementation gap, and future directions after the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: To obtain the solution, we present assumptions including second-order Taylor approximation and isotropic Hessian. Part of derivation (proof) is in the method section, while the remaining part is in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the pseudocode in Algorithm 1. The other experimental settings are the same as previous research, such as FAMO and FairGrad.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We put the link to code in the main paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We follow the same experiment settings as previous research, such as FAMO and FairGrad.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We repeat experiments for three times using different seeds and report the mean and standard deviation of metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the experiment section, we present the running time and the type of GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the social impact in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We wouldn't release any dataset. In future (after patent application is submitted), we would select restricted licenses when releasing the code or pre-trained model.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers of datasets and codes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: When submitting the paper, we wouldn't release new assets. In future, after the patent application, we would release the code together with documents.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for polishing writing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.