Graph Coloring for Multi-Task Learning

Santosh Patapati[1⋆] and Ian Noronha[2]

[1] Stony Brook University AI Innovation Institute, Stony Brook, NY 11794, USA
santosh.patapati@stonybrook.edu
[2] Purdue University, West Lafayette, IN 47907, USA
inoronha@purdue.edu

Abstract. When different objectives conflict with each other in multi-task learning, gradients begin to interfere and slow convergence, thereby potentially reducing the final model's performance. To address this, we introduce SON-GOKU, a scheduler that computes gradient interference, constructs an interference graph, and then applies greedy graph-coloring to partition tasks into groups that align well with each other. At each training step, only one group (color class) of tasks are activated, and the grouping partition is constantly recomputed as task relationships evolve throughout training. By ensuring that each mini-batch contains only tasks that pull the model in the same direction, our method improves the effectiveness of any underlying multi-task learning optimizer without additional tuning. Since tasks within these groups will update in compatible directions, multi-task learning will improve model performance rather than impede it. Empirical results on six different datasets show that this interference-aware graph-coloring approach consistently outperforms baselines and state-of-the-art multi-task optimizers. We provide extensive theory showing why grouping and sequential updates improve multi-task learning, with guarantees on descent, convergence, and the ability to accurately identify what tasks conflict or align.

Keywords: Multi-task learning · Graph coloring · Gradient interference

1 Introduction

Multi-task learning (MTL) trains a single model to solve several tasks simultaneously, sharing knowledge across them to learn more effectively [4, 8]. This allows models to generalize better and converge faster. However, a key issue known as negative transfer arises when tasks don't align very well with each other [40, 43]. When two tasks push the shared network in different directions their gradients clash, slowing or even reversing learning. Prior work addresses this issue primarily via (1) gradient manipulation, which reshapes task gradients to reduce conflicts, and (2) loss reweighting, which rescales task objectives to balance their influence. While effective in some specific settings, these strategies typically treat conflict

⋆ Corresponding author.
Implementation: https://anonymous.4open.science/r/SON-GOKU-Impl/

locally at the level of shared-parameter updates and often overlook the evolving global structure of interactions among tasks throughout training.

Some recent works focus on partitioning tasks into subsets (groups) and updating those groups separately. These approaches have been found to improve accuracy and training stability by forming groups with high measured affinity and then updating one group at a time [12, 18]. Grouping can outperform gradient manipulation and loss reweighting when tasks form clusters with aligned gradients, because each update then reduces direct clashes in the shared layers, lowers gradient variance within the step, and lets compatible tasks reinforce one another while conflicting tasks wait for their turn.

However, grouping methods often face a few key limitations: (1) many rely on dense pairwise affinities that grow noisy and costly as the number of tasks rises [12, 42, 47]; (2) others predetermine or rarely update groups, so they drift as task relations change [38, 51]; and (3) several use local heuristics that fail to enforce global compatibility or to specify how groups should rotate over time [29, 57].

We present **SON-GOKU** (**S**cheduling via **O**ptimal I**N**terference-aware **G**raph-C**O**loring for Tas**K** Grouping in M**U**ltitask Learning). We measure gradient interference, build a graph of tasks from those measurements, greedily color the graph to form non-conflicting compatible task groups, and update one color group per step during training. This design addresses the earlier issues. We estimate the interference graph from lightweight minibatch statistics and keep it sparse, which avoids noisy dense matrices and scales to many tasks. We recolor the graph at regular intervals so the groups track changing relations during training. Greedy graph coloring ensures we update only compatible tasks in each step, and the color order gives a simple way to cycle through the groups. Our proposed scheduler does not have to work in isolation and can function on top of existing loss-reweighting and gradient-manipulation MTL approaches.

In our theoretical analysis (Section 5) we show that, under standard conditions, SON-GOKU tends to group tasks whose gradients are, on average, aligned within each group, with high probability. We further show that, over a refresh window, sequentially updating these low-conflict groups yields *at least* as much expected descent as a single mixed update, and *strictly more* when between-group interference is sufficiently negative. We also prove that SON-GOKU preserves descent and reaches the usual non-convex SGD rate under mild assumptions, with only a small factor that depends on the within-group conflict level. In Supp. Materials Section D we discuss the scheduler's amortized time complexity and the tradeoffs it offers between speed and performance. We discuss ways in which practitioners can reduce its time complexity under certain conditions.

Empirical results from experiments demonstrate that SON-GOKU consistently improves outcomes compared to other MTL approaches, especially when SON-GOKU is coupled with existing approaches. Our contributions are as follows:

- We propose SON-GOKU, an interference-aware scheduler that measures cross-task gradient conflict, builds a conflict graph, colors it to form compatible

groups, and activates one group per step. It can be used on top of standard MTL optimizers.
- We provide theoretical analysis that offers guarantees on SON-GOKU's grouping, convergence, scheduling behavior, and more.
- Across six datasets, SON-GOKU improves over strong baselines and pairs well with methods like PCGrad, AdaTask, and GradNorm, delivering consistent gains [9, 55, 56].
- We perform an ablation study showing that dynamic recoloring and history-averaged conflict estimates are key contributors to performance.

2 Related Work

Prior work has identified the phenomenon of gradient interference in multi-task learning and explored several strategies to mitigate it. We group these strategies into four families: (1) *Tuned Loss Weighting*, (2) *Adaptive Loss Weighting*, (3) *Gradient-Level Conflict Mitigation*, and (4) *Empirical Task Grouping*. SON-GOKU falls into family (4).

Many MTL methods (especially earlier ones) adjust task influence by learning or adapting loss weights. Examples include uncertainty-based scaling [20], rate-based schemes such as DWA [28], and fast bilevel formulations like FAMO [26]. FAMO in particular is notable for its $\mathcal{O}(1)$ per-step time complexity. These approaches keep all tasks active each step while modulating relative magnitudes. A completely different approach, which emerged in 2018 with MGDA [40], focuses on updating shared-parameter *update directions* to mitigate interference [25]. Methods like PCGrad [56], CAGrad [27], and MGDA [40] modify the geometry of the shared update to reduce cross-task conflicts while still updating all tasks each step. A smaller body of work forms subsets of tasks to update together, using offline affinity estimation or training-dynamics signals [12, 42, 47, 51]. See Supp. Materials Section Q for additional analysis of non-conflict task grouping. Most recently, Selective Task Group Updates proposes online grouping with sequential updates, reporting that update order can influence task-specific learning [18]. We also distinguish SON-GOKU from GO4Align and EXTRA. GO4Align studies task imbalance and alignment, and EXTRA studies task tradeoffs, while SON-GOKU schedules tasks using measured gradient conflict. In contrast to these methods, SON-GOKU changes which tasks are updated together at each step rather than only changing task weights or analyzing tradeoffs [41, 59]. SON-GOKU differs in mechanism from existing approaches (Section 4). It complements loss reweighting and gradient surgery, and we provide explicit guarantees on descent, convergence, and graph partition recovery. An expanded discussion and commentary of related work is provided in Supp. Materials Section M.

3 Problem Setup

We formalize multi-task learning (MTL) [8] as optimizing a shared network while activating only a subset of tasks at each step. Each task contributes a loss whose

gradients may align or conflict. We quantify conflict using (the negative of) cosine similarity, embed tasks in a conflict graph, and later use that graph to derive a schedule (see Supp. Materials Section P for a unique, modular, formulation and results with alternative measures of affinity). This section fixes notation and states the optimization goal that the proposed approach addresses.

3.1 Data and Notation

Let $\mathcal{T} = \{T_1, \ldots, T_K\}$ be the set of tasks. The model has shared parameters $\theta \in \mathbb{R}^d$ and task-specific parameters $\phi_k \in \mathbb{R}^{d_k}$ for T_k. Each task draws examples (x, y_k) from a distribution $\mathcal{D}_k$ and defines a per-example loss $\ell_k(\theta, \phi_k; x, y_k)$. Its population loss is

$$L_k(\theta, \phi_k) := \mathbb{E}_{(x,y_k)\sim\mathcal{D}_k}\big[\ell_k(\theta, \phi_k; x, y_k)\big]. \tag{1}$$

We minimize the standard weighted MTL objective

$$F(\theta, \phi_1, \ldots, \phi_K) \;=\; \sum_{k=1}^{K} w_k\, L_k(\theta, \phi_k), \tag{2}$$

with nonnegative task weights w_k (default $w_k = 1$). Note that, for simplicity in later sections, we absorb w_k into the per-task gradient estimates. This is permissible since positive scalings do not change cosine signs or the induced conflict graph. We write $\mathcal{L}_k(\theta, \phi_k; \mathcal{B})$ for the corresponding weighted mini batch loss.

At step t, for any task k that is active we compute stochastic gradients on a mini-batch $\mathcal{B}_k^{(t)} \subset \mathcal{D}_k$:

$$g_k^{(t)} := \nabla_\theta \mathcal{L}_k(\theta_t, \phi_{k,t}; \mathcal{B}_k^{(t)}), \qquad h_k^{(t)} := \nabla_{\phi_k} \mathcal{L}_k(\theta_t, \phi_{k,t}; \mathcal{B}_k^{(t)}). \tag{3}$$

In our proposed method, we form exponential moving averages (EMA) of per-task gradients within a refresh window to stabilize cosine estimates so that they do not become stale (Sec. 4) [37].

Interference Coefficient We quantify pairwise interaction with the interference coefficient

$$\rho_{ij} \;=\; -\frac{\langle \tilde{g}_i, \tilde{g}_j \rangle}{\|\tilde{g}_i\|\,\|\tilde{g}_j\|}, \tag{4}$$

where $\tilde{g}_i$ and $\tilde{g}_j$ are the EMA-smoothed gradients at refresh. Positive ρ_{ij} indicates conflict (negative cosine). $\rho_{ij} \le 0$ indicates alignment or neutrality.

Conflict Graph Fix a tolerance $\tau \in (0, 1)$. The conflict graph is

$$G_\tau = (\mathcal{T}, E_\tau), \qquad E_\tau = \big\{(i, j) : \rho_{ij} > \tau\big\}. \tag{5}$$

Vertices are tasks. An edge between a pair means to not update that pair together. We will utilize G_τ for coloring and scheduling in Section 4

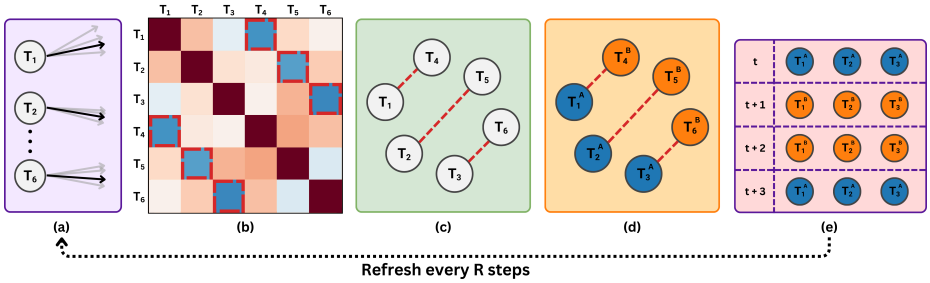

Fig. 1: Interference-aware scheduling pipeline: (a) For each task T_i (circles $T_1 \dots T_6$), we smooth recent per-step gradients with an Exponential Moving Average (EMA); (b) From these EMA vectors we compute the pairwise cosine matrix. In the figure, cells outlined with red dashes mark pairs with cosine $< -\tau$. These are flagged as conflicts; (c) We build the conflict graph whose nodes are tasks T_i and whose red dashed edges connect exactly those pairs identified in (b); (d) We apply greedy graph coloring so that no conflict edge lies within a color, producing low-conflict groups. In the example shown, we have two groups: A as blue and B as orange; (e) During training we activate one group per step. After every R steps (here, $R = 4$) we 'refresh' and run the pipeline again from step A, where we update the EMAs with the latest gradients.

3.2 Goal

At training step t we choose an active set $S_t \subseteq \mathcal{T}$ and update only those tasks:

$$\theta_{t+1} = \theta_t - \eta_t \sum_{k \in S_t} g_k^{(t)}, \qquad \phi_{k,t+1} = \begin{cases} \phi_{k,t} - \eta_t h_k^{(t)}, & k \in S_t, \\ \phi_{k,t}, & k \notin S_t. \end{cases} \tag{6}$$

The problem the scheduler addresses is to design the sequence $\{S_t\}_{t=1}^{T}$ so that: (1) every task is visited regularly; and (2) conflicting tasks seldom appear together. We instantiate this via greedy graph coloring in Section 4 and analyze the guarantees in Section 5.

4 Proposed Approach

We design an interference-aware scheduler that partitions tasks into low-conflict groups and activates exactly one group per optimization step. The procedure consists of four stages: (1) estimating pairwise interference, (2) building and coloring the conflict graph, (3) generating a periodic schedule, and (4) updating that schedule as training evolves. An overview of the scheduler is provided as Algorithm 1 in the Supplementary Materials. A visualization of SON-GOKU is provided in Figure 1 alongside a simple summary in the Figure caption.

4.1 Estimating Gradient Interference

Throughout this section, R denotes the refresh period. We use r for refresh rounds, t_r for the first training step of refresh round r, and m_r for the number

of color classes produced in that round. We use d_{sk} for the sketch width. We absorb task weights into per-task losses, so $g_k^{(t)}$ is the gradient of the weighted mini batch loss. Cosine calculations and graph construction are not impacted by applying positive scaling.

At step t and for every task T_k appearing in the current mini-batch we compute a task-specific stochastic gradient

$$g_k^{(t)} = \nabla_\theta \mathcal{L}_k\big(\theta_t, \phi_{k,t}; \mathcal{B}_k^{(t)}\big), \tag{7}$$

using an independent sub-batch $\mathcal{B}_k^{(t)} \subset \mathcal{D}_k$. We then update an exponential moving average

$$\tilde{g}_k^{(t)} = \beta\, \tilde{g}_k^{(t-1)} + (1-\beta)\, g_k^{(t)}, \qquad \beta \in [0,1), \tag{8}$$

which stabilizes cosine estimates while requiring only two buffers per task (current and previous). To estimate such cosines in practice, we employ low dimensional sketches of the EMA for each task, so the additional memory usage scales well [14, 54]. Whenever we refresh the schedule (every R steps) we form the pairwise interference matrix. The EMA gradients are used only to build the graph. The parameter updates use the current stochastic gradients.

$$\rho_{ij}^{(t)} \;=\; -\frac{\langle \tilde{g}_i^{(t)}, \tilde{g}_j^{(t)} \rangle}{\|\tilde{g}_i^{(t)}\|\, \|\tilde{g}_j^{(t)}\|}, \qquad i,j \in \{1,\dots,K\}. \tag{9}$$

Computing all $K(K-1)/2$ cosines via the Gram matrix in the d_{sk}-dimensional sketch space costs $O(Kdd_{\text{sk}} + K^2 d_{\text{sk}})$, namely $O(Kdd_{\text{sk}})$ to form the sketch M_f and $O(K^2 d_{\text{sk}})$ for the Gram product, where $d_{\text{sk}} \ll d$ is the sketch width (see Supp. Materials Section D.5.1). We also write $h_k^{(t)} = \nabla_{\phi_k} \mathcal{L}_k\big(\theta_t, \phi_{k,t}; \mathcal{B}_k^{(t)}\big)$ for the gradient with respect to the task-specific parameters ϕ_k.

4.2 Conflict Graph Construction

Given a tolerance $\tau \in (0,1)$, the conflict graph at update round r is

$$G_\tau^{(r)} = (V, E_\tau^{(r)}), \quad V = \{1,\dots,K\} E_\tau^{(r)} = \big\{(i,j) \;:\; \rho_{ij}^{(t_r)} > \tau\big\}. \tag{10}$$

To clarify, tasks are indexed by integers $1 \dots K$ in Equation 10. Edges connect tasks whose averaged gradients have cosine similarity less than $-\tau$. Intuitively, larger τ yields a sparser conflict graph, typically fewer colors (larger per-step groups), and more frequent updates per task. Smaller τ results in a denser graph, more colors (smaller per-step groups), and less frequent updates per task. This construction reflects optimization-time interference. $G_\tau^{(r)}$ is symmetric and undirected, derived from current gradient geometry to decide which tasks should not be updated together.

4.3 Partitioning via Greedy Graph Coloring

We apply the Welsh-Powell largest-first greedy heuristic [7, 52] to color $G_\tau^{(r)}$ and obtain color classes $C_1^{(r)}, \ldots, C_{m_r}^{(r)}$. Classical graph-theory results [10, 53] guarantee the heuristic uses no more than $\Delta + 1$ colors, where Δ is the maximum vertex degree. In practice Δ is small because many task pairs do not interfere, yielding concise schedules.

4.4 Schedule Generation and Execution

We create a periodic schedule of length m_r:

$$S_t \;=\; C^{(r)}_{\left(t \bmod m_r\right)+1}, \qquad t_r \le t < t_{r+1} = t_r + R. \tag{11}$$

Each training step activates exactly one color class; over one period every task in that class receives a gradient update, while conflicting tasks (edges in $E_\tau^{(r)}$) are guaranteed not to co-occur.

Minimum update frequency If the greedy coloring yields a singleton class for a rarely updated task, we increase its update frequency by duplicating it only into steps whose active color has no conflict edge to that task.

Warm-up and Annealing We start with $\tau = 1$ (no edges, full simultaneous training) for the first T_{warm} steps, then logarithmically anneal τ to a target value τ^* [15]. This mitigates noisy gradient signals early in training. Similarly, we can set the refresh period with a smaller R to adapt to changing gradients and increase it as training stabilizes (Supp. Materials Section O).

4.5 Time Complexity and Space Complexity

Using the sketched implementation described in Supp. Materials Section D, a single refresh of the SON-GOKU scheduler has time complexity $O(Kdd_{\text{sk}}+K^2d_{\text{sk}})$, where $d_{\text{sk}} \ll d$ is the sketch width. However, unlike many MTL approaches, our scheduler concentrates its extra work in occasional refreshes. This time complexity therefore becomes $O\Big(\frac{Kd_{\text{sk}}(d+K)}{R}\Big)$ amortized per training step where R is the refresh period (the number of training steps between conflict-graph rebuilds). For the small, fixed d_{sk} used in our experiments, this overhead still grows roughly quadratically in K but is independent of d up to the $O(Kdd_{\text{sk}})$ sketching term and shrinks linearly with the refresh period R. Similarly, SON-GOKU's persistent space complexity of $O(K^2)$ scales with K but not d, the number of model parameter dimensions, allowing it to maintain low memory usage even with large backbone models. We provide a full analysis of the time complexity in Supp. Materials Section D and discuss approaches to reducing time complexity under certain conditions in Supp. Materials Section D.5. See also Supp. Materials Section R for scaling behavior with larger backbones.

5 Theoretical Analysis

We discuss some of the main guarantees behind SON-GOKU. For a very brief overview: (1) Updating groups of tasks whose gradients are mostly low-conflict (no internal edges) reduces the objective on average and still achieves the usual $1/\sqrt{T}$ convergence rate; (2) Over a refresh window, scheduling several group updates can beat one mixed update that uses all tasks at once; and (3) With a small number of recent gradient measurements per task (via EMA) and a margin separating conflicts, the estimated conflict graph matches the ideal one, giving a short schedule where every task is updated at least once every $\Delta + 1$ steps (Δ is the maximum number of conflicts for any task). We provide expanded assumptions, definitions, proofs, reasoning, analysis, etc. in Supp. Materials Sections B–I (see also N, ¶–R).

5.1 Descent Preservation Within a Low-conflict Group

If the active set S_t at step t is τ-compatible, then the combined update formed from the weighted shared gradients is a descent direction with a quantitative lower bound:

$$\Big\| \sum_{k \in S_t} g_{k,t} \Big\|^2 \;\geq\; \Big(1 - \tau\,(|S_t| - 1)\Big) \sum_{k \in S_t} \|g_{k,t}\|^2 \tag{12}$$

Thus the step cannot flip to ascent whenever $\tau(|S_t| - 1) < 1$. This is proved by expanding the polarization identity and controlling cross terms under the τ-compatibility condition (see Supp. Materials Section E). Essentially, this means that SON-GOKU's per-step updates are safe when groups are low conflict. The aggregate direction keeps pointing downhill and the cancellation is quantitatively limited by τ and group size.

5.2 Nonconvex Convergence at the Standard Rate up to a Small Factor

Under standard smoothness and noise conditions (see Supp. Materials Section I) and with steps $\eta = c/\sqrt{T}$, SON-GOKU achieves the usual nonconvex SGD rate, with a mild $(1 + \tau)$ factor that reflects within-group conflict:

$$\min_{t<T} \mathbb{E}\,\|\nabla F(\theta_t)\|^2 \;\leq\; \frac{2(F_0 - F^\star)}{c\sqrt{T}}\,(1+\tau) \;+\; \frac{cL\sigma^2}{\sqrt{T}} \tag{13}$$

When $\tau = 0$, the constant matches the classical bound [6, 13]; as $\tau \to 1$, it at most doubles, matching the intuition that conflict can cancel up to half of the progress. This demonstrates that scheduling does not degrade asymptotic progress. SON-GOKU preserves the $1/\sqrt{T}$ decay of the gradient norm while controlling the constant through the compatibility threshold τ. In other words, we keep the standard rate of SGD and trade a small constant for reduced interference. We use SGD in the theory to isolate the scheduler. SON-GOKU selects active tasks before the optimizer step and does not reset optimizer states.

5.3 When Scheduled Groups Outperform a Single Mixed Update

We compare two ways to use the same gradients gathered at a refresh: a scheduled sequence of per-group steps (i.e., the scheduler used in SON-GOKU) versus a single aggregated step. Using a telescoping L-smooth bound and evaluating both trajectories at a common linearization (i.e., expanding F at the refresh start θ_{t_r} and applying the same first-order model with the same step size) the scheduled bound is never worse and is strictly better when cross-group interaction terms are sufficiently negative (so mixed updates would cancel progress).

Essentially, when different groups' gradients pull in opposing directions (so adding them together would cancel progress) the scheduler has an advantage. In that case, taking the updates one group at a time is provably better. Our theory guarantees a larger drop in the objective during that refresh than the one-shot step, even though both use the same step size and the same gradients. Under the PL condition, the scheduled path maintains the usual contraction factor and gains a nonnegative extra decrease term over the window [19].

5.4 Exact Recovery of the Population Conflict Graph and Task Partition

We show that, after observing gradients for only a modest number of steps, the scheduler can exactly reconstruct the true conflict relations among tasks by averaging recent gradients (EMA), computing pairwise cosines, thresholding at $-\tau$, and coloring the resulting graph. Under a separation margin γ around the threshold (tasks are meaningfully different), bounded noise, and bounded drift within each refresh window, the conflict graph estimated from finite data agrees, with high probability, with the ideal population conflict graph $G^\star\tau$ (defined from the pairwise cosines of the true mean gradients $\{\mu_i\}_{i=1}^K$ at the start of the refresh window). Equivalently, when the uniform cosine estimation error is below γ, we have $\widehat{G}_\tau = G^\star_\tau$ and the resulting grouping recovers the ground-truth task partition. This explains why the scheduler's group structure is trustworthy and ties the required number of recent gradient measurements per task to interpretable quantities such as noise level, margin, and the number of tasks. For example, an effective sample size of $n_{\text{eff}} \gtrsim \frac{\sigma^2}{m_0^2\gamma^2}\log\big(K/\delta\big)$ suffices in our analysis.

5.5 Scheduling Properties with Few Groups and Bounded Staleness

Welsh-Powell greedy coloring uses at most $\Delta+1$ colors on a graph whose maximum degree is Δ [5]. Running the colors in a fixed cycle means each task is updated at least once every $m \le \Delta + 1$ steps. Equivalently, no task waits more than Δ steps between updates (bounded staleness).

This means that the schedule length is controlled by the worst conflict degree Δ rather than by the total number of tasks K. This results in two important benefits: (1) a minimum update-frequency guarantee, since every task receives an update at least once per cycle of length $\le \Delta + 1$; and (2) compatibility with standard bounded-delay conditions used in analyses of asynchronous SGD

(e.g., [23, 34, 36]), with delay parameter at most Δ. When $\Delta \ll K$, we achieve both low interference (few conflicts per step) and low staleness (short update gaps).

6 Experimental Setup

6.1 Datasets

We evaluate across six benchmarks spanning vision, multimodal, and time-series [1, 22, 45, 50]. For each dataset we specify a small set of primary tasks and add positive and negative auxiliaries to stress interference. Architectures are standard backbones (e.g., ResNet-18 for image tasks [16], CNN/BiLSTM for time-series) with task-specific heads. Full dataset and task definitions, auxiliary construction, and architecture details (including preprocessing and head designs) are provided in the Supplementary Materials under section J and Table 4. We provide additional experiments with varying backbones in Supp. Materials Section R.

6.2 Baseline and State-of-the-Art Comparisons

We compare against loss-weighting (Uniform, GradNorm, AdaTask), multi-objective (MGDA, Nash-MTL, FairGrad), projection/surgery (PCGrad, CA-Grad), and fast adaptive weighting (FAMO) [3, 9, 26, 35, 39, 55, 56]. We provide short method notes in Supp. Materials Section K and discuss these approaches in Section 2.

6.3 Scheduler Extension Models

In addition to standalone models, we also evaluate combinations of the scheduler with existing approaches. These combinations use SON-GOKU only to select tasks before the optimizer update.

1. *SON-GOKU + AdaTask.* Combines our interference-aware task selection with AdaTask's dynamic loss weighting, applying adaptive weights only to scheduler-selected tasks.
2. *SON-GOKU + GradNorm Warm Start.* Initializes training with GradNorm for stable gradient magnitudes, then transitions to our scheduler after 3 epochs.
3. *SON-GOKU + PCGrad.* Applied PCGrad's gradient projection specifically to tasks selected by our scheduler, providing fine-grained conflict resolution within τ-compatible groups.

Single-Step Conflict Estimation Here, we set the history length to $H = 1$, so every recoloring step relies on only the most recent mini-batch gradients to estimate interference. Without aggregation over many past steps, the conflict graph should become highly noisy, causing unstable task groupings from one update window to the next. This variant tests the importance of historical conflict statistics in the scheduler.

Table 1: Performance of Evaluated Approaches Across Datasets. *DM* represents Density-Matched ablation variants

Model	Accuracy (%) ↑			F&B		HEALTH		NYUv2		
	CIFAR-10	AV-MNIST	MM-IMDb	Acc. (%) ↑	MAE ↓	Acc. (%) ↑	MAE ↓	Angle Error ↓	Seg. mIoU ↑	Depth RMSE ↓
Uniform	55 ±2.2	63 ±1.5	56 ±2.8	45 ±2.4	0.57 ±0.030	52 ±2.0	0.54 ±0.024	21.6 ±0.27	0.059 ±0.003	0.73 ±0.018
GradNorm	61 ±1.6	65 ±1.1	58 ±2.0	47 ±2.3	0.57 ±0.020	53 ±2.1	0.52 ±0.019	21.4 ±0.23	0.054 ±0.004	0.65 ±0.016
MGDA	59 ±2.9	62 ±1.7	56 ±3.3	44 ±3.0	0.57 ±0.036	53 ±2.5	0.53 ±0.030	21.8 ±0.33	0.063 ±0.005	0.75 ±0.024
PCGrad	61 ±1.9	65 ±1.3	58 ±2.3	50 ±2.1	0.55 ±0.024	58 ±2.0	0.48 ±0.021	20.9 ±0.24	0.070 ±0.004	0.69 ±0.013
CAGrad	59 ±2.0	62 ±1.1	57 ±2.5	46 ±2.5	0.58 ±0.031	53 ±1.9	0.52 ±0.024	21.9 ±0.29	0.065 ±0.004	0.73 ±0.018
AdaTask	63 ±1.5	67 ±0.9	59 ±1.9	47 ±1.9	0.59 ±0.026	55 ±2.2	0.52 ±0.024	20.3 ±0.23	0.069 ±0.004	0.65 ±0.015
FAMO	64 ±1.2	70 ±1.0	61 ±1.6	52 ±2.0	0.53 ±0.021	60 ±1.8	0.49 ±0.018	19.9 ±0.19	0.074 ±0.003	0.63 ±0.012
FairGrad	62 ±1.8	66 ±1.3	59 ±2.5	52 ±2.5	0.54 ±0.026	60 ±2.0	0.47 ±0.022	20.7 ±0.27	0.072 ±0.004	0.67 ±0.015
Nash-MTL	63 ±1.9	66 ±1.2	60 ±2.1	52 ±2.3	0.54 ±0.024	60 ±2.3	0.47 ±0.023	20.6 ±0.24	0.073 ±0.004	0.67 ±0.013
Static One-Shot	61 ±2.0	66 ±1.1	58 ±2.6	48 ±2.3	0.56 ±0.027	54 ±2.1	0.51 ±0.025	20.5 ±0.25	0.071 ±0.004	0.65 ±0.016
Single-Step	40 ±4.2	59 ±2.4	20 ±5.4	42 ±3.9	0.60 ±0.041	47 ±3.5	0.55 ±0.034	26.4 ±0.55	0.042 ±0.006	0.81 ±0.029
SON-GOKU (Threshold, DM)	63	68	59	49	0.55	56	0.51	20.6	0.071	0.61
SON-GOKU (kNN-Symm.)	60	65	55	46	0.57	52	0.53	22.1	0.066	0.70
SON-GOKU (kNN-Symm., DM)	61	66	57	47	0.56	54	0.52	21.4	0.068	0.66
SON-GOKU (Signed-only)	56	63	52	43	0.60	50	0.56	24.0	0.053	0.76
SON-GOKU (Signed-only, DM)	58	64	54	45	0.59	52	0.54	23.0	0.056	0.73
SON-GOKU (Quantile)	64	68	60	50	0.54	57	0.50	20.3	0.072	0.60
SON-GOKU (Quantile, DM)	65	69	61	51	0.53	58	0.50	20.0	0.072	**0.59**
SON-GOKU + GradNorm	62 ±1.4	69 ±1.0	59 ±1.7	51 ±1.8	0.53 ±0.022	59 ±1.7	0.49 ±0.018	**19.6** ±0.19	0.073 ±0.003	0.64 ±0.011
SON-GOKU + AdaTask	**67** ±1.2	**71** ±0.9	**63** ±1.6	52 ±1.7	0.53 ±0.021	59 ±1.8	0.48 ±0.017	20.1 ±0.20	0.068 ±0.004	0.67 ±0.013
SON-GOKU + PCGrad	65 ±1.3	70 ±0.9	60 ±1.8	**54** ±2.0	**0.52** ±0.024	**62** ±1.6	**0.45** ±0.020	19.7 ±0.18	**0.076** ±0.003	0.62 ±0.010
SON-GOKU	65 ±1.5	69 ±1.0	61 ±1.8	51 ±1.9	0.53 ±0.023	58 ±1.7	0.50 ±0.018	19.8 ±0.20	0.073 ±0.004	**0.59** ±0.012

7 Results and Discussion

Results for all models across every experiment are depicted in Table 1. All metrics are held-out test results under identical training setups and architectures. Across ten metrics on six datasets, our conflict-aware schedulers consistently match or exceed all baseline methods.

7.1 Overall Performance Improvements

Overall, the conflict-aware approaches improve over the uniform baseline by 10%-20% on CIFAR-10 and by 7% on MM-IMDb, indicating that grouping tasks according to measured interference is more effective than treating all tasks equally at every update. On NYUv2, we see similar improvements across all the metrics. These results suggest that the scheduler's graph coloring cleanly separates high-conflict tasks, preserving the projection or LR-balancing advantages (stemming from PCGrad's gradient projection and AdaTask's learning-rate adaptation, respectively) while removing residual interference (see Supp. Materials Section S for grouping patterns at training time and more analyses). As we evaluated across diverse tasks and datasets, our results also demonstrate clear improvements in generalization.

7.2 Ablation Study on Scheduler Design

We evaluate nine controlled ablations of six types: (i) **Static One-Shot Coloring**, which runs greedy graph coloring once at the start of training and then freezes

the groups, testing dependence on *dynamic* recoloring as gradients change; (ii) **Single-Step Conflict Estimation**, which sets the history length to $H = 1$ so each recoloring uses only the most recent mini batch, testing the importance of averaging conflict statistics over time; (iii) **Threshold Graph (baseline)**, which connects tasks i and j when the smoothed cosine $\hat{s}_{ij}(t)$ falls below a global threshold $-\tau(t)$; (iv) **kNN-Symmetric Graph**, which connects each task to its m most conflicting neighbors and then symmetrizes the edges, enforcing roughly fixed degree per task and comparing local degree control against the global threshold rule; (v) **Signed-Only Graph**, which adds an edge only if $\hat{s}_{ij}(t) < 0$, yielding a very sparse graph and ignoring moderate (but potentially harmful) conflicts; and (vi) **Quantile Threshold Graph**, which at each refresh sets $\tau(t)$ so that only the worst $p\%$ of cosine values are treated as conflicting, keeping edge density approximately stable and testing an adaptive cutoff versus a fixed global threshold. We evaluate each graph rule under two settings. In the *fixed* τ setting, all rules share the same $\tau(t)$ schedule used in the main experiments. In the *density-matched* setting, we adjust the hyperparameters of each rule so that all graphs have approximately the same edge density at each refresh. This isolates the effect of which pairs are marked as conflicting, rather than how many edges are present. We go into much further detail regarding the ablation in Supp. Materials Section K.3.

These ablations directly test the assumptions behind SON-GOKU. Static One-Shot, which freezes groups, consistently underperforms the full scheduler on most metrics, indicating that task relations change enough during training that dynamic recoloring is needed to maintain τ-compatibility as gradients drift (Sections 5.1–5.2). Single-Step, which uses $H = 1$, is clearly worse across datasets, matching our claim that batch cosines are too noisy [21]. Instead, averaging conflict statistics over short history windows provides the clean information needed for accurate graph recovery (Section 5.4). Among graph constructions, simple threshold and quantile rules (and their density-matched variants) perform similarly well, suggesting that any approach that reliably isolates the worst conflicting pairs is sufficient. In contrast, Signed-Only and kNN-Symmetric, which ignore conflict magnitude or have purely local degree control, degrade performance more noticeably, especially on NYUv2 and the tabular benchmarks. Overall, the best performing configurations are precisely those that match the descent and recovery conditions analyzed in Sections 5.1–5.2 and 5.4.

7.3 Additional Analysis

Optimizer-Task Alignment Interestingly, we observe that AdaTask-based approaches tend to be the best on classification tasks (CIFAR-10, AV-MNIST, MM-IMDb) while PCGrad-based approaches tend to be the best on tasks that model regression (NYUv2).

We believe that this stems from unique differences in the features of classification and regression-based models. For example, cross-entropy gradients near decision boundaries tend to be bursty and high in variance [17, 24, 44, 46]. By

scaling each task's step size according to its running gradient norm, AdaTask smooths out these spikes.

On the other hand, we believe that PCGrad under the scheduler performs particularly well on regression and dense-prediction tasks as their tasks tend to generate smooth, large-magnitude gradients whose directions change gradually. PCGrad removes only the small component of the gradient that conflicts across tasks, preserving the main descent direction while reducing interference.

Synergy Between Scheduling and Baselines We believe that the superior results found in the combinations of the scheduler and baseline models can be traced to the way scheduling and optimization reinforce one another.

First, greedy graph coloring partitions tasks into τ-compatible groups, segregating tasks with highly divergent gradients. This yields a guaranteed lower bound on descent (Proposition 6 under Supplementary Materials), directly improving optimization efficiency.

Within each low-conflict group, the optimizer can do its job under more ideal conditions. PCGrad can remove the remaining minor conflicting components, preserving the majority of the descent direction. AdaTask can adjust each task's learning rate without being impacted by large adversarial gradients.

This $\Delta + 1$ color bound ensures that every task is scheduled at least once per period. This prevents tasks from being essentially starved of updates.

Finally, by computing interference over a window, the scheduler smooths out gradient fluctuations [31]. This prevents the erratic schedule changes that projection-only grouping methods have been shown to face [43, 56, 58], thereby better stabilizing convergence.

Optimization Structure and Held-out Performance While our guarantees in Section 5 and Supp. Materials Sections B–F are stated in optimization terms, they help explain the held out gains by increasing gradient coherence and limiting destructive interference. Section 5.1 shows that the aggregated group gradient remains aligned with descent and that intra-group gradient conflict is explicitly limited by τ and $|S_t|$. Section 5.3 then compares two ways to apply the same gradients during a refresh, either a single mixed update or a scheduled sequence of group updates. Together, these analyses imply that each step in SON-GOKU provides more informative signals and less interference, or, equivalently, a higher gradient-to-noise ratio [11, 30, 32, 46, 48]. Building on this, Section 5.4 shows that SON-GOKU's estimated conflict graph recovers the population structure with high probability, so the schedule repeatedly updates clusters of related tasks rather than conflicting tasks. By enforcing positive affinity within groups, SON-GOKU is able to train related tasks together. This enables effective sharing of model parameters across different tasks, reducing the complexity of the model and increasing sample efficiency [2, 8, 49]. With this alongside a high gradient-to-noise signal ratio, SON-GOKU can improve performance across many different datasets, domains, and distributions and can perform well even under non-ideal conditions (e.g., noisy labels, class or task imbalance, distribution shift, etc.) [33]. Our

Table 2: Wall-clock time (seconds ± standard deviation) vs. number of tasks K.

Method (R if applicable)	K=3	K=6	K=16	K=40
Uniform	0.2656 ± 0.1201	0.3240 ± 0.0629	0.3798 ± 0.1050	0.4054 ± 0.1190
GradNorm	5.4714 ± 0.7137	5.1201 ± 0.6112	4.9042 ± 0.5869	4.7372 ± 0.9286
AdaTask	2.1816 ± 0.0934	2.1032 ± 0.1012	2.2853 ± 0.0718	2.2278 ± 0.1370
PCGrad	3.6212 ± 0.3517	23.1266 ± 0.8773	176.7566 ± 2.8171	1127.1337 ± 34.2603
MGDA	97.1081 ± 5.4645	121.4371 ± 9.0923	132.4913 ± 3.1752	134.0878 ± 2.2621
FAMO	2.0725 ± 0.2073	1.9980 ± 0.1998	2.1710 ± 0.2171	2.1164 ± 0.2116
FairGrad	3.8020 ± 0.5703	15.2079 ± 2.2812	108.1450 ± 16.2218	675.9065 ± 101.3860
Nash-MTL	5.7030 ± 1.1406	22.8118 ± 4.5624	162.2176 ± 32.4435	1013.8598 ± 202.7720
SON-GOKU ($R = 32$)	1.9896 ± 0.3651	3.3202 ± 0.5745	6.0897 ± 0.9425	12.1432 ± 1.2044
SON-GOKU + AdaTask ($R = 32$)	3.7718 ± 0.9654	5.0511 ± 0.6531	7.5903 ± 1.1920	14.5182 ± 2.0660
SON-GOKU + GradNorm ($R = 32$)	7.0202 ± 1.0711	8.1661 ± 0.9355	10.7227 ± 2.2088	16.5760 ± 1.8418
SON-GOKU + PCGrad ($R = 32$)	1.9834 ± 0.3586	3.4971 ± 0.3840	6.1395 ± 0.9425	10.9097 ± 1.5263

ablation results (Table 1) demonstrate that variants without dynamic recoloring or history averaging perform worse, indicating accurate and low-conflict grouping is essential.

7.4 Speed and Tradeoffs

SON-GOKU has a time complexity of $O\big(Kd_{\text{sk}}(d+K)/R\big)$ (Section 4.5) amortized per training step (Section 4.5). Table 2 shows near-linear growth over this range of K at R=32, reflecting sparsity in the graphs and batched cosine computation. SON-GOKU's time rises from around 2 seconds ($K = 3$) to 12 seconds ($K = 40$), remaining far below methods that perform heavy conflict handling. For example, PCGrad, FairGrad, and Nash-MTL increase steeply with K. In contrast, FAMO and AdaTask are among the fastest and largely flat with K, as expected from their constant overhead.

SON-GOKU is also memory efficient, with an only incremental memory footprint that scales with the number of tasks K, not the parameter dimension d. The scheduler's peak memory during a refresh step is $O(K^2+Kd_{\text{sk}})$ and the persistent state between refreshes is $O(K^2)$ (see Supp. Materials Section N for further theoretical and experimental analysis). By contrast, methods that retain K full gradients require $O(Kd)$ additional memory. This implies that, on larger backbones (high d), SON-GOKU's memory overhead is modest and grows mainly with the task count K, rather than with model size.

These contrasts demonstrate the tradeoffs between speed and fidelity to task interference. Faster methods like FAMO minimize overhead, while methods that model conflicts can improve accuracy. These tradeoffs have to be assessed on a case-by-case basis, based on values that factor into each approach's time complexity and the importance of training speed versus performance.

8 Conclusion

We introduced SON-GOKU, an interference-aware scheduler that estimates cross-task alignment, builds a sparse conflict graph, and greedily colors it to activate one low-conflict group per step. Formally, we provide rigorous theoretical guarantees that justify the design and effectiveness of the scheduler. Empirically, across six benchmarks, SON-GOKU improves over strong baselines and recent approaches. It complements optimizers like PCGrad and AdaTask, indicating that scheduling and gradient shaping are synergistic. By modeling task interactions with a conflict graph and schedule, SON-GOKU offers a simple, scalable, and theory-backed mechanism for robust multitask training.

Acknowledgements

This work was supported in part by the Lambda Research Grant, which provided GPU cloud compute resources through Lambda Cloud. We also gratefully acknowledge the Google TPU Research Cloud (TRC) program for providing Cloud TPU compute resources used in our experiments.

References

1. Arevalo, J., Solorio, T., Montes-y Gómez, M., González, F.A.: Gated multimodal units for information fusion. arXiv preprint arXiv:1702.01992 (2017)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Advances in Neural Information Processing Systems. vol. 19, pp. 41–48 (2007)
3. Ban, H., Ji, K.: Fair resource allocation in multi-task learning. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, JMLR.org (2024)
4. Baxter, J.: A model of inductive bias learning. J. Artif. Int. Res. **12**(1), 149–198 (Mar 2000)
5. Bonamy, M., Kelly, T., Nelson, P., Postle, L.: Bounding χ by a fraction of δ for graphs without large cliques (2018), https://arxiv.org/abs/1803.01051, accessed July 31 2025
6. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM review **60**(2), 223–311 (2018)
7. Brélaz, D.: New methods to color the vertices of a graph. Commun. ACM **22**(4), 251–256 (Apr 1979). https://doi.org/10.1145/359094.359101, https://doi.org/10.1145/359094.359101, accessed 27 June 2026
8. Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (Jul 1997). https://doi.org/10.1023/A:1007379606734, https://doi.org/10.1023/A:1007379606734, accessed July 5 2025
9. Chen, Z., Badrinarayanan, V., Lee, C., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: International Conference on Machine Learning. pp. 794–803 (2018), https://arxiv.org/abs/1711.02257, accessed February 2 2026
10. Diestel, R.: Graph Theory. Springer, 5th edn. (2017)

11. Fan, C., Chen, W., Tian, J., Li, Y., He, H., Jin, Y.: Maxgnr: A dynamic weight strategy via maximizing gradient-to-noise ratio for multi-task learning. arXiv preprint arXiv:2302.09352 (2023)
12. Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., Finn, C.: Efficiently identifying task groupings for multi-task learning (2021), https://arxiv.org/abs/2109.04617, accessed October 30 2025
13. Ghadimi, S., Lan, G.: Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization **23**(4), 2341–2368 (2013). https://doi.org/10.1137/120880811
14. Ghashami, M., Liberty, E., Phillips, J.M., Woodruff, D.P.: Frequent directions: Simple and deterministic matrix sketching. SIAM Journal on Computing **45**(5), 1762–1792 (2016). https://doi.org/10.1137/15M1009718
15. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hoffer, E., Hubara, I., Soudry, D.: Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. In: Advances in Neural Information Processing Systems. vol. 30, pp. 1731–1741 (2017)
18. Jeong, W., Yoon, K.J.: Selective task group updates for multi-task optimization (2025)
19. Karimi, H., Nutini, J., Schmidt, M.: Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 795–811. Springer (2016)
20. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7482–7491 (2018). https://doi.org/10.1109/CVPR.2018.00781
21. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=H1oyRlYgg
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Lian, X., Huang, Y., Li, Y., Liu, J.: Asynchronous parallel stochastic gradient for nonconvex optimization. Advances in neural information processing systems **28** (2015)
24. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2999–3007 (2017)
25. Lin, X., Zhen, H.L., Li, Z., Zhang, Q.F., Kwong, S.: Pareto multi-task learning. Advances in neural information processing systems **32** (2019)
26. Liu, B., Feng, Y., Stone, P., Liu, Q.: Famo: Fast adaptive multitask optimization. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 57226–57243. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/b2fe1ee8d936ac08dd26f2ff58986c8f-Paper-Conference.pdf, accessed October 29 2025

27. Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. In: Advances in Neural Information Processing Systems. vol. 34, pp. 12345–12355 (2021)
28. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1871–1880 (2019)
29. Malhotra, A., Vatsa, M., Singh, R.: Dropped scheduled task: Mitigating negative transfer in multi-task learning using dynamic task dropping. Transactions on Machine Learning Research (2022)
30. Mandt, S., Hoffman, M.D., Blei, D.M.: A variational analysis of stochastic gradient algorithms. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. p. 354–363. ICML'16, JMLR.org (2016)
31. Mandt, S., Hoffman, M.D., Blei, D.M.: Stochastic gradient descent as approximate bayesian inference. Journal of Machine Learning Research **18**(134), 1–35 (2017)
32. McCandlish, S., Kaplan, J., Amodei, D., Team, O.D.: An empirical model of large-batch training. arXiv preprint arXiv:1812.06162 (2018)
33. Michalkiewicz, M., Faraki, M., Yu, X., Chandraker, M., Baktashmotlagh, M.: Domain generalization guided by gradient signal to noise ratio of parameters. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6177–6188 (2023)
34. Mitliagkas, I., Zhang, C., Hadjis, S., Ré, C.: Asynchrony begets momentum, with an application to deep learning. In: 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton). p. 997–1004. IEEE Press (2016). https://doi.org/10.1109/ALLERTON.2016.7852343, https://doi.org/10.1109/ALLERTON.2016.7852343, accessed 27 June 2026
35. Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., Fetaya, E.: Multi-task learning as a bargaining game. arXiv preprint arXiv:2202.01017 (2022)
36. Niu, F., Recht, B., Ré, C., Wright, S.J.: Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In: Advances in Neural Information Processing Systems. vol. 24, pp. 693–701 (2011)
37. Polyak, B.T., Juditsky, A.B.: Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization **30**(4), 838–855 (1992). https://doi.org/10.1137/0330046, https://doi.org/10.1137/0330046, accessed 27 June 2026
38. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017), https://arxiv.org/abs/1706.05098, accessed December 4 2025
39. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 525–536. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
40. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Advances in Neural Information Processing Systems. vol. 31, pp. 525–536 (2018)
41. Shen, J., Wang, C., Xiao, Z., Noord, N.V., Worring, M.: GO4align: Group optimization for multi-task alignment. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=8vCs5U9Hbt, accessed August 7 2025
42. Sherif, A., Abid, A., Elattar, M., ElHelw, M.: Stg-mtl: scalable task grouping for multi-task learning using data maps. Machine Learning: Science and Technology

5(2), 025068 (Jun 2024). https://doi.org/10.1088/2632-2153/ad4e04, http://dx.doi.org/10.1088/2632-2153/ad4e04, accessed September 22 2025
43. Shi, G., Li, Q., Zhang, W., Chen, J., Wu, X.: Recon: Reducing conflicting gradients from the root for multi-task learning. In: ICLR 2023 Workshop on Multi-Task Learning (2023)
44. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 761–769 (2016)
45. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European conference on computer vision. pp. 746–760. Springer (2012)
46. Smith, S.L., Le, Q.V.: A bayesian perspective on generalization and stochastic gradient descent. arXiv preprint arXiv:1710.06451 (2017)
47. Standley, T., Zamir, A.R., Chen, D., Guibas, L., Malik, J., Savarese, S.: Which tasks should be learned together in multi-task learning? In: Proceedings of the 37th International Conference on Machine Learning (ICML). pp. 9120–9132 (2020)
48. Sun, Z., Sun, Y., Yang, L., Lu, S., Mei, J., Zhao, W., Hu, Y.: Unleashing the power of gradient signal-to-noise ratio for zero-shot nas. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5763–5773 (2023)
49. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3614–3633 (2022). https://doi.org/10.1109/TPAMI.2021.3054719
50. Vielzeuf, V., Lechervy, A., Pateux, S., Jurie, F.: Centralnet: a multilayer approach for multimodal fusion. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
51. Wang, C., Pan, X., Yu, T.: Towards principled task grouping for multi-task learning. arXiv preprint arXiv:2402.15328 (2024)
52. Welsh, D.J.A., Powell, M.B.: An upper bound for the chromatic number of a graph and its application to timetabling problems. The Computer Journal **10**(1), 85–86 (01 1967). https://doi.org/10.1093/comjnl/10.1.85, https://doi.org/10.1093/comjnl/10.1.85, accessed August 24 2025
53. West, D.B.: Introduction to Graph Theory. Prentice Hall, 2nd edn. (2000)
54. Woodruff, D.P., et al.: Sketching as a tool for numerical linear algebra. Foundations and Trends® in Theoretical Computer Science **10**(1–2), 1–157 (2014)
55. Yang, E., Pan, J., Wang, X., Yu, H., Shen, L., Chen, X., Xiao, L., Jiang, J., Guo, G.: Adatask: A task-aware adaptive learning rate approach to multi-task learning. In: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI) (2023), https://arxiv.org/abs/2211.15055, accessed July 8 2025
56. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: Advances in Neural Information Processing Systems. vol. 33, pp. 18524–18536 (2020)
57. Zhang, Y., Yang, Q.: An overview of multi-task learning. National Science Review **5**(1), 30–43 (2018)
58. Zhang, Z., Shen, J., Cao, C., Dai, G., Zhou, S., Zhang, Q., Zhang, S., Shutova, E.: Proactive gradient conflict mitigation in multi-task learning: A sparse training perspective. arXiv preprint arXiv:2411.18615 (2024), https://arxiv.org/abs/2411.18615, accessed September 14 2025
59. Zhou, Z., Meng, Z., Wu, P., Zhao, P., Miao, C.: Exploring tradeoffs through mode connectivity for multi-task learning. In: The Thirty-ninth Annual Conference on

Neural Information Processing Systems (2026), https://openreview.net/forum?id=4ULtNYHc5T, accessed February 22 2026