Gradient Similarity Surgery in Multi-Task Deep Learning

Thomas Borsani[1 [0009−0007−9541−890X]] (✉), Andrea Rosani[1 [0009−0008−2622−6776]], Giuseppe Nicosia[2 [0000−0002−0650−3157]], and Giuseppe Di Fatta[1 [0000−0003−3096−2844]]

[1] Free University of Bozen-Bolzano, Bozen-Bolzano, Italy
{tborsani, Andrea.Rosani, Giuseppe.DiFatta}@unibz.it
[2] University of Catania, Catania, Italy
nicosia@dmi.unict.it

Abstract. The multi-task learning (MTL) paradigm aims to simultaneously learn multiple tasks within a single model capturing higher-level, more general hidden patterns that are shared by the tasks. In deep learning, a significant challenge in the backpropagation training process is the design of advanced optimisers to improve the convergence speed and stability of the gradient descent learning rule. In particular, in multi-task deep learning ($MTDL$) the multitude of tasks may generate potentially *conflicting gradients* that would hinder the concurrent convergence of the diverse loss functions. This challenge arises when the gradients of the task objectives have either different magnitudes or opposite directions, causing one or a few to dominate or to interfere with each other, thus degrading the training process. Gradient surgery methods address the problem explicitly dealing with conflicting gradients by adjusting the overall gradient trajectory. This work introduces a novel gradient surgery method, the Similarity-Aware Momentum Gradient Surgery (SAM-GS), which provides an effective and scalable approach based on a gradient magnitude similarity measure to guide the optimisation process. The SAM-GS surgery adopts gradient equalisation and modulation of the first-order momentum. A series of experimental tests have shown the effectiveness of SAM-GS on synthetic problems and MTL benchmarks. Gradient magnitude similarity plays a crucial role in *regularising gradient aggregation* in $MTDL$ for the optimisation of the learning process. Code is available at https://unibzmlgroup.github.io/SAMGS/

Keywords: Multi-Task Deep Learning · Gradient Descent Optimisation · Gradient Surgery · Gradient Aggregation · Conflicting Gradients .

1 Introduction

In the multi-task learning (MTL) paradigm [1] a model is trained on multiple tasks simultaneously, leveraging a shared internal representation to improve generalisation and efficiency. While training a model for a single task leverages on patterns in the data, training on multiple tasks also leverages on patterns in the

tasks. MTL exploits task similarities to enhance performance, particularly when tasks share some underlying features. Utilising a shared representation for many tasks allows to improve model generalisation by capturing features that are more resilient to noise compared to a single-task approach. This concurrent learning process acts as a regularisation mechanism, reducing bias and strengthening the robustness of the model. Additionally, this approach is advantageous when data availability is particularly heterogeneous across tasks, as it enables the aggregation of data from many tasks to improve overall learning. Moreover, MTL can lead to a reduction in computational costs, training and inference time, but this depends on the specific implementation and task relationships.

The MTL paradigm has been successfully applied to many problems across various domains, including Natural Language Processing [26,34], Computer Vision [4,33], Healthcare and Medical Imaging [14,15], Fraud Detection and Finance [25]. These applications demonstrate how MTL can improve generalisation, reduce data requirements, and enhance model efficiency across diverse real-world problems. Nevertheless, there are challenges to effectively training MTL models, particularly in selecting and combining tasks, as different tasks may not always align seamlessly to produce better solutions [32].

Recent research has evidenced that one of the primary challenges for the optimisation of MTL models is the aggregation of the different gradients associated to the task-specific loss functions [37]. Typically, the task gradients are aggregated using the arithmetic mean. Indeed, it has been shown that this approach can lead to suboptimal solutions [32,37]. The underlying cause have been identified in the challenges arising from the aggregation of conflicting task gradients, i.e. gradients with opposite directions (*angle-based conflicting gradients*) and gradients dominating the aggregation (*magnitude conflicting gradients*) [37].

Current solutions to address the problem of conflicting gradients can be categorised into three sub-groups. *Task Similarity* methods focus on the selection of tasks that do not cause gradient conflicts [12,39]. *Loss Balancing* methods focus on static or dynamic weighting algorithms to weight the different loss functions [19,2], and *Gradient Surgery* methods seek to mitigate gradient conflicts by applying heuristics that modify the gradient descent learning rule to reduce their impact [37,20,24].

However, methods of *Task Similarity* tend to be computationally inefficient and limit MTL applicability, serving primarily to avoid the problem rather than addressing it to optimise the potential benefits offered by MTL models. *Loss Balancing* methods, while effective and more efficient than task similarity methods in addressing the problem [19], still ignore its underlying causes. *Gradient Surgery* methods tackle gradient conflicts directly and have been shown to be among the most effective strategies to optimise MTL models [21,24,27]. Most of these methods, however, apply the procedure indiscriminately, overlooking the proper identification of gradient conflicts, which can lead to a deterioration of the original gradient-based learning process. Additionally, some of these methodologies excessively level out the relative contributions of the tasks to the overall

gradient, and inevitably miss out the inherent advantage of MTL, where tasks may provide complementary contributions in the shared representation.

To address the issue of conflicting gradients while accounting for the varying nature of task loss functions, we introduce a novel gradient surgery method, the Similarity-Aware Momentum Gradient Surgery (SAM-GS). This method dynamically adapts the gradient descent optimisation process based on the task gradient magnitude similarity. The proposed approach applies a conservative learning when gradients are dissimilar and accelerates learning when they exhibit high similarity. SAM-GS integrates gradient equalisation within conflicting scenarios and incorporates a gradient momentum, whose influence is adaptively modulated based on the task gradient similarity. Comparative experimental results demonstrate that this adaptive strategy enhances stability and efficiency in learning dynamics, yielding superior performance across diverse MTL benchmarks.

Key contributions of the proposed SAM-GS method are as follows:

- *SAM-GS Optimisation*: Introduction of a gradient similarity measure to selectively adjust gradient magnitudes, enhancing the learning process.
- *Momentum-Based Regularisation*: Integration of gradient momentum into gradient surgery, introducing a new regularisation for conflicting gradients, improving the optimisation dynamics.
- *Empirical Validation*: Analysis on synthetic problems and evaluation on four standard MTL benchmarks, achieving comparable or improving state-of-the-art (SOTA) performance over existing methods.

The remainder of the paper is organised as follows. In Section 2, we present the problem of conflicting gradients in $MTDL$ and the solution offered by gradient surgery methods. In Section 3, we discuss related work in terms of the three approaches, task similarity, load balancing and gradient surgery, to compare and contrast them. In Section 4, the proposed SAM-GS method is introduced and its main algorithm described. In Section 5, we present an experimental and comparative analysis of the proposed method with respect to other gradient surgery methods. Section 6 provides the main conclusions and indicates some areas of improvement.

2 MTL Optimisation

In this section, we introduce the definition of the multi-task learning paradigm, discuss the specific challenge referred to as *conflicting gradients* in deep learning models, and provide an overview of gradient surgery methods.

The MTL paradigm aims to optimise a single model $\theta \in \mathbb{R}^m$ for $K \geq 2$ numbers of tasks simultaneously. In general, the objective is to minimise the sum of the task-specific loss functions $\mathcal{L}_i(\theta) : \mathbb{R}^m \to \mathbb{R}_+$

$$\arg\min_{\theta \in \mathbb{R}^m} \left\{ \mathcal{L}_{mtl}(\theta) := \sum_{i=1}^{K} \mathcal{L}_i(\theta) \right\} \tag{1}$$

The training of a MTL model through direct optimisation of the Equation (1) may yield to sub-optimal solutions, characterised by under-optimised tasks [37]. More specifically, MTL can be framed as a multi-objective optimisation problem [7], where optimising Equation (1) may result in solutions that are not Pareto-efficient. In deep network models the literature has identified gradient conflicts as one of the primary causes of this sub-optimisation issue [37].

2.1 Conflicting Gradients in $MTDL$

In training deep learning models on multiple tasks simultaneously, the issue of conflicting gradients arises when different tasks produce gradients that interfere with each other, leading to inefficient or suboptimal learning. This detrimental interference hinders the performance of the model across tasks.

Two main types of conflicting gradients can be identified, respectively, caused by the relative direction of the task gradient vectors and by their different magnitudes.

Angle-Based Gradient Conflict. Let $g_i, g_j \in \mathbb{R}^d$ be the gradient vectors associated with two different tasks i and j. We define an *angle-based gradient conflict* as occurring when the angle ϕ_{ij}, in Equation (2), between them is greater than 90, which corresponds to a negative cosine similarity. In this situation, the vector sum reduces the net effective learning step, slowing convergence [37].

$$\cos(\phi_{ij}) = \frac{g_i \cdot g_j}{\|g_i\|\|g_j\|} < 0. \tag{2}$$

In this scenario, the least critical case occurs when the gradients from different tasks are nearly orthogonal to each other. This still results in inefficient learning since updates get diluted rather than reinforcing progress in the common direction. The most critical case arises when the gradients from different tasks are perfectly opposite to each other, resulting in a zero vector and effectively preventing learning.

Magnitude Gradient Conflict. Let $g_i, g_j \in \mathbb{R}^d$ be the gradient vectors associated with two different tasks i and j. We quantify the *magnitude gradient conflict* by means of the magnitude similarity defined in Equation (3):

$$\psi(g_i, g_j) = \frac{2\|g_i\|_2\|g_j\|_2}{\|g_i\|_2^2 + \|g_j\|_2^2}. \tag{3}$$

A magnitude gradient conflict occurs when the gradients associated with different tasks have significantly varying magnitudes. This imbalance can cause the model to prioritise certain tasks over others, leading to suboptimal performance.

In contrast to the *angle-based gradient conflict*, where it is clearly defined when two gradients are in conflict (i.e., negative cosine similarity), the detection of magnitude-based gradient conflicts is less straightforward. Dissimilarities in task gradient magnitudes may not be due to actual conflicts but to the lack of

loss normalisation or to local topological differences in loss functions across the tasks.

2.2 Gradient Surgery Methods

Gradient surgery methods provide a heuristic aggregation function over the task gradient vectors to compute the overall gradient driving the weight update rule. The surgery function is aimed at optimising all tasks effectively by limiting the effect of gradient conflicts. We introduce a generic task gradient aggregation function, which determines how gradients from different tasks are combined according to the surgery method. The gradient of the total loss with respect to the weight matrix θ at layer is l is: $\nabla_{\theta^{(l)}}\mathcal{L} = s\left(\nabla_{\theta^{(l)}}\mathcal{L}_1, \nabla_{\theta^{(l)}}\mathcal{L}_2, \ldots, \nabla_{\theta^{(l)}}\mathcal{L}_K\right)$, where $s(\cdot)$ is a task gradient aggregation function that determines how the individual task gradients contribute to the overall optimisation.

3 Related Works

Existing solutions to deal with gradients conflicts can been categorised in three main groups, as follows.

Task Similarity. The optimisation via Task Similarity methods aims to group tasks that can be learned synergistically, thereby improving overall model performance. It is also possible that the best solution does not involve using one MTL model to solve K tasks, but rather employing K single-task models, which may lead to better outcomes [12,39,29,32,28]. In this approach, gradient conflicts are avoided by selecting a suitable combination of tasks that do not present conflicts.

Loss Balancing. Loss Balancing methods relies on weighting the different loss functions of the tasks involved in the combination. Various methodologies have been proposed to determine the optimal weights for different tasks. The method **UW** [5] leverages the homoscedastic uncertainty of each task to determine the weights , while **DWA** [22] utilises rate of change of task-specific loss functions. **GradNorm** [2] modulates weights based on the magnitude of the gradient. In contrast to these approaches, **RLW** [18] assigns random weights. Additionally, **FAMO** [19] learns the weights based on the quality of the loss updates. These methods mitigates gradient conflicts by preventing any single task from dominating the training process.

Gradient Surgery. These methods aim to enhance convergence in MTL by appropriately weighting the gradient components of different tasks. They focus on introducing heuristics to adjust the combination of gradient vectors, thereby influencing the optimisation process dynamics to resolve the conflicts and guiding the model more effectively through the loss landscape. Approaches like **Nash-MTL** [24] utilise game theory concepts, particularly the Nash Bargaining Solution, to equilibrate task gradients. Instead, **MGDA** [10,8] for MTL seeks

a direction that minimises all objectives simultaneously, in line with the multi-objective Karush–Kuhn–Tucker (KKT) [17] conditions. These methods are computationally intensive but have proven to be effective.

Alternative approaches aim to mitigate gradient conflicts. **GDOD** [9] decomposes task gradients into shared and conflicting components, updating only the shared ones. **PCGrad** [37] reduces conflicts among task gradients by decorrelating them while **CAGrad** [20] seeks a conflict-averse gradient path to minimise task interference. **GradDrop** [3] ensures consistency in gradient signs across tasks. In addition, **IMTL** [21] identifies a gradient path in which cosine similarities among task gradients remain consistent, and **Aligned-MTL** [27] mitigates conflicts by aligning the principal components of the gradient matrix. These methods compete effectively with the more complex Nash-MTL [24] showing better performance maintaining low computational overhead.

4 Similarity-Aware Momentum Gradient Surgery

Similarity-Aware Momentum Gradient Surgery (SAM-GS) is a gradient surgery method that leverages a measure of the magnitude similarity of the task gradients to detect and address conflicts during the learning process.

Here, we first present the intuition behind the approach with an example with four scenarios, and then we introduce the SAM-GS algorithm.

The proposed approach focuses solely on *magnitude gradient conflicts*, which are arguably critical to effective $MTDL$ optimisation. *Angle-based gradient conflicts* are intentionally disregarded, as they only impact convergence speed.

The core difficulty of $MTDL$, compared to STL, stems from the presence of *magnitude gradient conflicts*, which are unique to $MTDL$ and the primary source of task-specific conflicts [11]. In contrast, *Angle-based gradient conflicts* are more characteristic of inter-sample variation typically address with mini-batch gradient descent.

Let us consider why *angle-based gradient conflicts* can slow the convergence of the learning process while *magnitude gradient conflicts* can significantly hinder the overall optimisation preventing the convergence of some tasks. When adding two vectors g_i and g_j of similar magnitude ($|g_i| \simeq |g_j|$) at an angle α greater than 90°, the magnitude of the sum is reduced by a factor proportional to $cos(\alpha)$ compared to adding them when they are collinear, as shown in Figure 1a and 1b. In the worst case, when $\alpha = 180$, the two vectors are in exactly opposite directions, and their magnitudes cancel out. However, this extreme case is rather unlikely. Although reduced in magnitude, the vector sum still contains useful information about the direction of optimisation for the gradient descent algorithm. Hence, to enhance the magnitude of the resulting sum vector by means of the momentum with no need to detect this type of conflict explicitly.

However, when one of the task gradients is overly greater than the others ($|g_i| \gg |g_j|$) the overall sum of the gradients will result in a direction dominated by that single vector. This case can be quite detrimental as only one task will benefit from the learning process, as shown in Figure 1c. In this case, we

introduce a conflict detection mechanism and a procedure to equalise the task gradients before their aggregation.

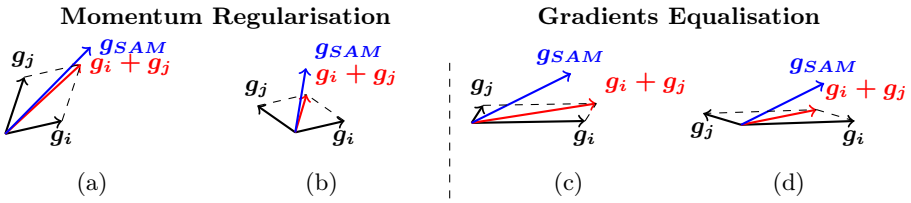


Fig. 1: Illustration of four scenarios for two task gradients, $\boldsymbol{g_i}$ and $\boldsymbol{g_j}$, the standard overall gradient is denoted as $\boldsymbol{g_i + g_j}$ and the overall gradient of SAM-GS is denote as $\boldsymbol{g_{SAM}}$. (a) Ideal case: Gradients have similar magnitudes, and the angle between them is less than 90°, indicating no conflict. (b) *angle-based gradient conflict*: The angle between gradients exceeds 90°, diminishing the effectiveness of their combination. (c) *magnitude-based gradient conflict*: One gradient dominates, leading to an imbalanced gradient update. (d) Both conflicts: A combination of angle- and magnitude-based gradient conflicts, where both the directional misalignment and magnitude disparity hinder effective gradient aggregation.

As illustrated in Figure 1, therefore, *magnitude gradient conflicts* have the potential to steer the optimisation process away from a fair convergence of all tasks, while *angle-based gradient conflicts* only influence the pace of convergence.

For this reason, SAM-GS ignores *angle-based gradient conflicts* and introduces two mechanisms: momentum regularisation and gradients equalisation. In particular, the momentum is modulated by the magnitude similarity, and the gradients equalisation is triggered by the detection of *magnitude gradient conflicts* by means of the magnitude similarity.

In cases where task gradients exhibit significantly different magnitudes, our approach equalises their magnitudes to compute a balanced direction not dominated by one task. The resulting sum vector is then scaled by the average magnitude to prevent the occurrence of near-zero gradients.

SAM-GS follows a general structure that is similar to ADABelief [40]. SAM-GS is specifically designed for multi-gradient optimisation, while ADABelief is applied to a single gradient (STL). ADABelief adopts a regularisation of the momentum that is based on the gradient, whereas SAM-GS applies a regularisation technique based on a gradient similarity measure.

Accordingly, SAM-GS is presented in Algorithm 1, where γ is a learnable hyperparameter to set the threshold on the gradient similarity to detect *magnitude gradient conflicts*. Let the model parameter vector at step t be represented by θ_t, it follows that for each of the $K \geq 2$ tasks, there exist a differentiable loss function, $\{l_i\}_{i=1}^{K}$. Consequently, for each task, the gradients $g_k = \nabla_\theta \mathcal{L}_k$ can be computed. The average magnitude similarity of the gradients, denoted as Ψ_t, is computed from the gradient magnitude similarities of Equation (3). Further-

more, we indicate the momentum with $m_{k,t}$, which is the exponential moving average (EMA) of $g_{k,t}$, and with h_t the EMA of $(1-\Psi_t)^2$ (similarity momentum coefficient) with β_1 and β_2 the smoothing parameters and $\hat{\cdot}$ represents the bias-corrected value of the respective quantity.

Algorithm 1 Similarity-Aware Momentum Gradient Surgery

Hyperparameters: $\beta_1 \leftarrow 0.9, \beta_2 \leftarrow 0.99, \gamma \leftarrow 0.1$
Initialise: $\theta_0, m_0 \leftarrow 0, h_0 \leftarrow 0, t \leftarrow 0, \epsilon \leftarrow 1e-8$
repeat
  $t \leftarrow t+1$
  $\boldsymbol{g}_k \leftarrow \nabla_{\theta_t} \mathcal{L}_k, \forall k$
  $\Psi = \frac{1}{K^2} \sum_{i,j} \psi(g_i, g_j)$
  $\boldsymbol{m}_{k,t} \leftarrow \beta_1 \boldsymbol{m}_{k,t-1} + (1-\beta_1)\boldsymbol{g}_k, \forall k$
  $h_t \leftarrow \beta_2 h_{t-1} + (1-\beta_2)(1-\Psi)^2 + \epsilon$
  $\widehat{\boldsymbol{m}}_{k,t} \leftarrow \frac{\boldsymbol{m}_{k,t}}{1-\beta_1^t}$, $\widehat{h}_t \leftarrow \frac{h_t}{1-\beta_2^t}$
  if $\Psi < \gamma$ **then**
    $\boldsymbol{w}_k = \frac{\|\boldsymbol{g}_k\|_2}{\|\boldsymbol{g}_k\|_2} \boldsymbol{g}_k, \forall k$
  else
    $\boldsymbol{w}_k = \frac{|\widehat{\boldsymbol{m}}_{k,t}|}{\sqrt{\widehat{h}_t + \epsilon}}, \forall k$
  end if
  Update: $\theta_t = \theta_{t-1} - \alpha \sum_{k=1}^{K} \boldsymbol{w}_k \odot \boldsymbol{g}_k$
until convergence

The proposed SAM-GS approach mitigates gradient dominance by adopting cautious updates with smaller step sizes. Conversely, when gradients are well-balanced, it leverages the momentum to accelerate learning and compensate for prior conservative updates. h_t acts as a regularisation term, where, if the gradients are dissimilar, the momentum is trusted less. Conversely, when the gradients exhibit good magnitude similarity, the momentum retains its full potential. The parameter γ plays a crucial role in determining the threshold at which gradients are considered well-balanced. We provide an ablation study on this parameter in section 5.2.

5 Computational Experiments and Comparisons

We conduct a series of experiments to empirically demonstrate the effectiveness of SAM-GS compared to other methods on synthetic problems and on common multi-task supervised benchmarks. Two variants of a synthetic problem based on two parameters are used to highlight the effect of gradient conflicts and how different methods fair under such conditions. The benchmarks based on real-world problems allow a comparative performance analysis of the proposed method against many state-of-the-art optimisation methods for MTL. An ablation study of SAM-GS hyperparameter γ allows to investigate its impact on

the performance of the method. In the following, each experimental setup is described and the results are presented.

5.1 Synthetic Problem

To illustrate the gradient surgery problem in a simplified setting, we adopt the 2D multi-task optimisation problem proposed in Nash-MTL [24]. This problem provides a controlled environment for the study of conflicting gradient across tasks, highlighting the challenges of multi-task optimisation. In addition, we introduce a novel variant of that problem with a similar loss landscape structure, featuring two global minima, providing a different problem setting to analyse the impact of multiple optima on optimisation dynamics.

Two-task problem with one global optimum. The synthetic problem proposed in [24] provides a useful toy problem to investigate and visualise the behaviour of multi-task optimisation methods in a complex yet comprehensible loss landscape. The problem consists of two loss functions with two parameters, and the objective is to minimise both using an MTL optimisation approach; a detailed formulation is reported in [20].

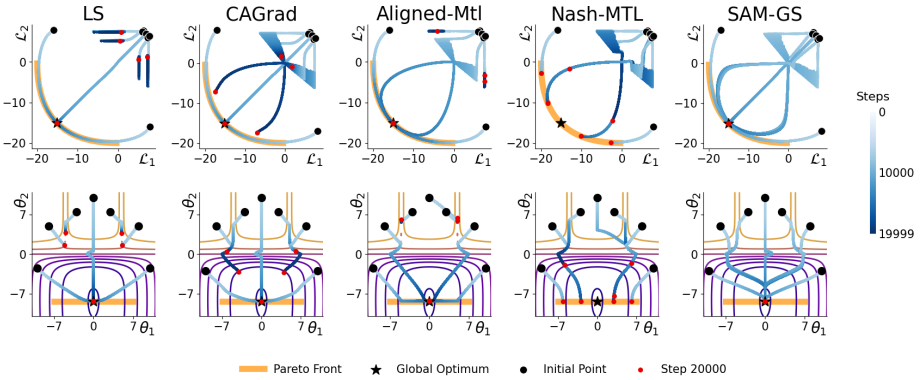
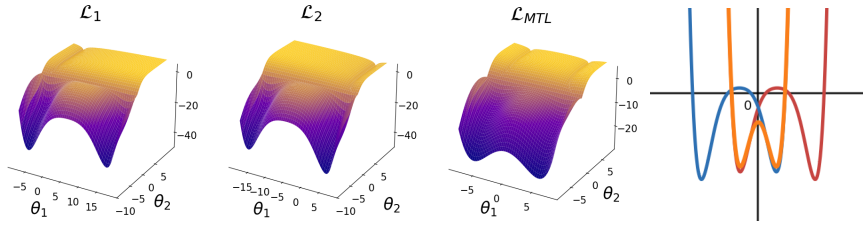


Fig. 2: Trajectories for different methods starting from 7 different initial points: Linear Sum (LS) approach using Adam [16], Nash-MTL [24], CAGrad [20], Aligned-MTL [27], and SAM-GS, from the starting points to the global optimum at the centre of the Pareto front in the loss space (top row) and parameter space (bottom row). The red dots show the end state of the trajectory after 20,000 iterations.

In the experimental results shown in Figure 2, indicate that the proposed approach exhibits behaviour comparable to CAGrad [20]. The maximum number of steps is set to 20,000: SAM-GS converges within 18,000 steps, and the simulation was run for 10% more steps to ensure a good comparison. Our method



(a) (b) (c) (d)

Fig. 3: Illustration of the multi-task optimisation problem ($\mathcal{L}_{MTL}$) computed as the sum of $\mathcal{L}_1$ and $\mathcal{L}_2$. In panel (d), the loss functions, $\mathcal{L}_1$ and $\mathcal{L}_2$ are displayed in red and blue, respectively, as a function of θ_1 given $\theta_2 = -5$, and $\mathcal{L}_{MTL}$ is displayed in orange.

is the only one that consistently reaches the global optimum from all the considered starting points. This superior performance highlights the effectiveness of SAM-GS in navigating complex loss scenarios over existing methods.

Two-task problem with two global optima. We propose a novel inspired by Nash-MTL [24], where we introduce two distinct global optima to evaluate the MTL optimisation methods in a multi-optima scenario. In this setup, two loss functions, each dependent on two parameters, exhibit one global optimum and one local optimum. The combination of these functions forms a multi-task optimisation problem with two global optima corresponding to the two local optima of the single task problems, as illustrated in Figure 3. This problem setup is interesting because the MTL optima correspond to the single-task local minima, thus challenging the optimisation process. Additionally, this setup presents a saddle point, which is absent in the first synthetic problem, introducing a further complexity. The complete formulation of this setup is detailed in the supplementary material.

We compare our SAM-GS with LS using Adam [16], Nash-MTL [24], CAGrad [20], and Aligned-MTL [27] across six different initialisation points, running the algorithm for a maximum of 20,000 steps.

As shown in Figure 4, SAM-GS is the method that reaches one of the two global optima for most of the considered initial points within the maximum number of iterations. The ability of the method to consistently and efficiently identify a global optimum across different initialisations highlights its potential for solving complex multi-task optimisation problems with multiple optima, ensuring faster and more reliable convergence than existing approaches.

5.2 Performance Analysis

We tested the effectiveness of SAM-GS on three different multi-task supervised benchmarks, which have been used by various competitive optimisation methods

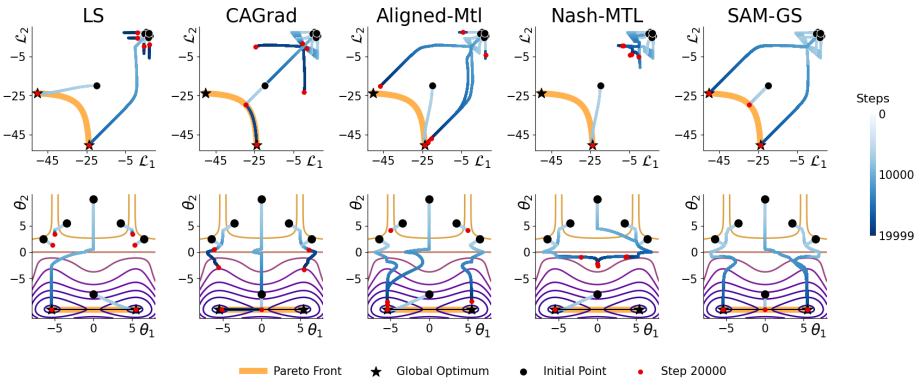


Fig. 4: Trajectories for different methods in the second synthetic problem: Linear Sum (LS) approach using Adam [16], Nash-MTL [24], CAGrad [20], Aligned-MTL [27], and SAM-GS, starting from six initial points and converging to one global optima at the extremes of the Pareto front in the loss space (top row) and parameter space (bottom row).

[19,21,24,27], CelebA [23] (40 tasks), NYU-v2 [30] (3 tasks) and CityScapes [6] (2 tasks). We compare SAM-GS against 14 different optimisation methods for multi-task learning. These include loss balancing methods such as UW [5], DWA [22], GradNorm [2], and RGW [18], as well as FAMO [19]. Additionally, we evaluate gradient surgery methods, including PCGrad [37], CAGrad [20], GradDrop [3],MGDA [8], IMTL [21], Nash-MTL [24] and Aligned-MTL [27].

In the remainder of this section we present the evaluation metrics used for the comparative analysis, the results on three benchmarks and, finally, the ablation study on SAM-GS hyperparameter.

Evaluation Metrics. To evaluate the performance of the optimisation methods, we use the Mean Ranking (**MR**) and the $\mathbf{\Delta m}\%$ metrics, similar to Nash-MTL [24]. The MR metric is the average rank of each method across tasks, where an MR of 1 indicates that the method ranks first on all tasks. The $\mathbf{\Delta m}\%$ metric, defined in Equation (4), quantifies the percentage improvement or degradation in performance of a method compared to the baseline single-task models.

$$\Delta m\% = \frac{1}{K}\sum_{k=1}^{K}(-1)^{\nu_k}\frac{m_{mtl,k} - m_{stl,k}}{m_{stl,k}} \cdot 100 \tag{4}$$

Here, $m_{mtl,k}$ and $m_{stl,k}$ represent the performance metrics for the MTL optimisation method and single-task models, respectively, for task k. The binary indicator ν_k is set to 1 when a higher value of m indicates better performance (e.g., accuracy), and 0 when a lower value is preferable (e.g., error).

CityScapes (2 tasks). The CityScapes dataset [6] contains 5,000 street-level RGBD images with per-pixel annotations across 19 semantic segmentation categories, grouped into 7 main categories. We adopt a similar experimental setup used in Nash-MTL [24], training a single Multi-Task Attention Network (MTAN) [22] model to simultaneously perform depth estimation and semantic segmentation. We identify that the best hyperparameter for SAM-GS are, $\beta_1 = 0.9$, $\beta_2 = 0.9$, $\gamma = 0.9$. Results in Table 1 shows that in this settings SAM-GS it is competitive with other methodology, but not superior in term of $\Delta m\%$. Some methods (e.g. UW [5]) have strictly better $\Delta m\%$ by excelling in one task; our approach has more balanced competitive performance across all tasks. The superior performance of Aligned-MTL [27], which focuses only on *angle-based gradient conflicts*, indicates that in this dataset inter-sample conflicts are more relevant, as also shown in [11]. This may explain the limitations of the proposed approach for this dataset.

Table 1: CityScapes results

	Segmentation		Depth			
	mIoU ↑	PixAcc ↑	AbsErr ↓	RelErr ↓	**MR** ↓	$\Delta m\%$ ↓
STL	74.01	93.16	0.0125	27.77		
LS	71.0	91.7	0.0161	33.8	11.8	14.1
SI	71.0	91.7	0.0161	33.8	11.8	14.1
RLW	74.6	93.4	0.0158	47.8	11.0	24.4
DWA	75.2	93.5	0.016	44.4	8.5	21.4
UW	72.0	92.8	0.014	**30.1**	7.75	5.89
MGDA	68.8	91.5	0.0309	33.5	12.5	44.1
PCGrad	75.1	93.5	0.0154	42.1	9.12	18.3
GradNorm	73.7	93.0	0.0124	34.1	7.75	5.63
GradDrop	75.3	93.5	0.0157	47.5	7.75	23.7
CAGrad	75.2	93.5	0.0141	37.6	7.88	11.6
IMTL-G	75.3	93.5	0.0135	38.4	6	11.1
Nash-MTL	75.4	**93.7**	**0.0129**	35.0	3.75	6.82
FAMO	74.5	93.3	0.0145	32.6	7.50	8.13
Aligned-MTL	**75.8**	93.7	0.0133	32.66	**2**	**5.27**
SAM-GS	75.2	93.5	0.0136	33.1	5.00	6.41

NYU-V2 (3 tasks). The NYU-v2 dataset [30] comprises 1,449 RGBD images of indoor scenes, with dense pixel-level annotations across 13 classes. We follow a similar experimental setup to Nash-MTL [24], training a single MTAN [22] model to perform depth estimation, image segmentation, and surface normal prediction. We identify the following hyperparameters for SAM-GS: $\beta_1 = 0.9$, $\beta_2 = 0.9$, $\gamma = 0.9$. The results in Table 2 show superior performance of SAM-GS, compared to other methods, in cases with more than two tasks.

CelebA (40 tasks). The CelebA dataset [23] is a collection of 200,000 facial images of 10,000 distinct celebrities, with 40 binary annotations of facial attributes for each image. We use the experimental setup outlined in FAMO

Table 2: NYU-V2 results

	Segmentation		Depth		Surface Normal						
	mIoU↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Angle Dist ↓		Within t° ↑				
					Mean	Median	11.25	22.5	30	**MR** ↓	$\Delta m\%$ ↓
STL	38.3	63.76	0.6754	0.278	25.01	19.21	30.14	57.2	69.15		
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.5	61.08	11.4	5.59
SI	38.45	64.27	0.5354	0.2201	27.6	23.37	22.53	48.57	62.32	10.3	4.39
RLW	37.17	63.77	0.5759	0.241	28.27	24.18	22.26	47.05	60.62	13.8	7.78
DWA	39.11	65.31	0.551	0.2285	27.61	23.18	24.17	50.18	62.39	10.2	3.57
UW	36.87	63.17	0.5446	0.226	27.04	22.61	23.54	49.05	63.65	10.0	4.05
MGDA	30.47	59.9	0.607	0.2555	24.88	19.45	29.18	56.88	69.36	7.4	1.38
PCGRAD	38.06	64.64	0.555	0.2325	27.41	22.8	23.86	49.83	63.14	10.6	3.97
GradNorm	20.09	64.64	0.7200	0.2800	**24.83**	**18.86**	**30.81**	**57.94**	**69.73**	7.2	7.22
GradDrop	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	9.6	3.58
CAGrad	39.79	65.49	0.5486	0.225	26.31	21.58	25.61	52.36	65.58	7.1	0.2
IMTL-G	39.35	65.6	0.5426	0.2256	26.02	21.19	26.2	53.13	66.24	6.3	-0.76
Nash-MTL	40.13	65.93	0.5261	0.2171	25.26	20.08	28.4	55.47	68.15	4.2	-4.04
FAMO	38.88	64.9	0.5474	0.2194	25.06	19.57	29.21	56.61	68.98	4.8	-4.1
Aligned-MTL	**40.82**	66.33	0.5300	0.2200	25.19	19.71	28.88	56.23	68.54	3.6	-4.93
SAM-GS	40.79	**66.46**	**0.5251**	**0.2169**	25.03	19.65	29.26	56.35	68.78	**2.4**	**-5.3**

[19], training a CNN model to perform 40 binary classification tasks. The hyperparameter search on the validation data identifies the following as the best hyperparameters for SAM-GS: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\gamma = 0.9$. The results in Table 3 show a superior performance of SAM-GS in handling 40 different tasks concurrently.

Ablation study on γ. In this section we provide a systematic study over the values of the similarity threshold γ.

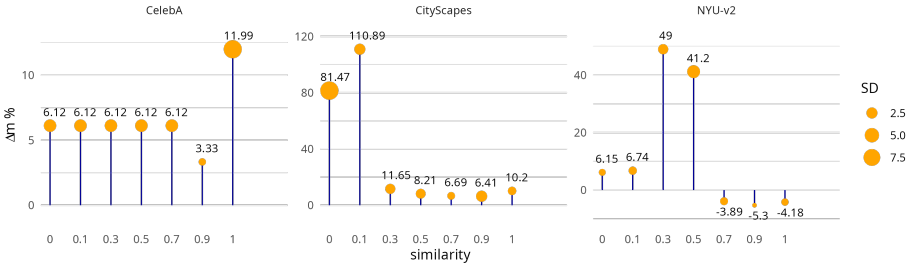


Fig. 5: Ablation study over γ. The plot shows the performance, in terms of $\Delta m\%$, of SAM-GS across three supervised learning settings with γ values of $\{0, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$, including standard deviation (SD)

Figure 5 highlights the critical role of γ in model performance. Extreme settings ($\gamma = 0$ or $\gamma = 1$), which make the algorithm to rely exclusively on either

Table 3: CelebA results.

Method	$\Delta m\% \downarrow$
LS	6.28
SI	7.83
RLW	5.22
DWA	6.95
UW	5.78
MGDA	10.93
PCGrad	6.65
GradDrop	7.80
CAGrad	6.20
IMTL-G	4.67
Nash-MTL	4.97
FAMO	4.72
Aligned-MTL	4.58
SAM-GS	**3.33**

Table 4: Reinforcement learning (MT10).

Method	**Success** $(mean \pm stderr)$
STL SAC	0.90 ± 0.032
MTL SAC	0.49 ± 0.073
MTL SAC + TE	0.54 ± 0.047
MH SAC	0.61 ± 0.036
SM	0.73 ± 0.043
CARE	0.84 ± 0.051
PCGrad	0.72 ± 0.022
CAGrad	0.83 ± 0.045
Nash-MTL	0.91 ± 0.031
Aligned-MTL	$\mathbf{0.97 \pm 0.045}$
FAMO	0.83 ± 0.05
SAM-GS	0.91 ± 0.018

the equalisation or the momentum component of SAM-GS, yield suboptimal results. On the other hand, intermediate values of γ, with a general trend towards higher settings, yield preferable results.

5.3 MTDL Reinforcement Learning (10 tasks)

Finally, we tested SAM-GS on a multi-task reinforcement learning (RL) problem against the most relevant MTL methods specifically designed for RL problems and a selection of the most recent gradient surgery methods. Specifically, we applied a variation of the SAM-GS method to the MetaWorld [38] MT10 benchmark, which comprises 10 distinct robot manipulation tasks with various reward functions. The variation concerns the computation of Ψ_t; we found that using $\Psi_t = \min \psi(g_i, g_j)$ led to improved results compared to averaging in a multi-task reinforcement learning problem. The experimental setting is similar to the one used in CAGrad [20], using Soft Actor-Critic (SAC) [13] as a baseline, trained with various gradient manipulation methods [37,20,24,27,19]. We also evaluate MTL-RL [31] approaches, including MTL SAC, Multi-task SAC with task encoder (MTL SAC + TE) [35], Multi-headed SAC (MH SAC) [35], Soft Modularization (SM) [36], and CARE [31]. We identify the hyperparameters for SAM-GS: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\gamma = 0.9$. The results presented in Table 4 indicate that SAM-GS achieves performance levels on par with Nash-MTL [24], while surpassing STL baseline, FAMO [19], CAGrad [20], and the standard gradient descent baseline method.

6 Conclusions

In multi-task deep learning training a single model on many tasks can be affected by potentially conflicting task gradients that would hinder the concurrent convergence of the diverse loss functions. In this study, the importance of the gradient magnitude similarity for the effective overall optimisation of the model has been studied and highlighted. As a result, a novel gradient surgery method, the Similarity-Aware Momentum Gradient Surgery (SAM-GS), has been proposed. SAM-GS is based on a measure of the task gradient magnitude similarity and used to control and guide two mechanisms: a momentum-based regularisation and a remedy for gradient magnitude conflicts. An extensive evaluation has demonstrated that SAM-GS effectively addresses a range of challenges with respect to task gradient conflicts and outperforms previous optimisation methods in two synthetic problems, several benchmarks from real-world computer vision applications, and a benchmark for reinforcement learning tasks. Future work may include a theoretical analysis of convergence to provide optimisation guarantees. Moreover, a direction for further improvements is the analysis of the current limitations to address strict stationary states such as saddle points, where task gradients have very similar magnitude and opposite directions.

Acknowledgments. We would like to thank the anonymous reviewers for their thorough reviews and insightful comments.

References

1. Caruana, R.: Multitask learning. Machine Learning **28**, 41–75 (1997)
2. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 794–803 (2018)
3. Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., Anguelov, D.: Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In: Advances in Neural Information Processing Systems. vol. 33, pp. 2039–2050 (2020)
4. Choi, W., Shin, M., Lee, H., Cho, J., Park, J., Im, S.: Multi-task learning for real-time autonomous driving leveraging task-adaptive attention generator. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 14732–14739 (2024)
5. Cipolla, R., Gal, Y., Kendall, A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7482–7491 (2018)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Di Fatta, G., Nicosia, G., Ojha, V., Pardalos, P.: Encyclopedia of Optimization, chap. Multi-Task Deep Learning as Multi-Objective Optimization (2020)

8. Dong, D., Wu, H., He, W., Yu, D., Wang, H.: Multi-task learning for multiple language translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1723–1732 (2015)
9. Dong, X., Wu, R., Xiong, C., Li, H., Cheng, L., He, Y., Qian, S., Cao, J., Mo, L.: Gdod: Effective gradient descent using orthogonal decomposition for multi-task learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 386–395 (2022)
10. Désidéri, J.A.: Multiple-gradient descent algorithm (mgda) for multiobjective optimization. Comptes Rendus Mathematique **350**(5), 313–318 (2012)
11. Elich, C., Kirchdorfer, L., Köhler, J.M., Schott, L.: Examining common paradigms in multi-task learning. In: Pattern Recognition. pp. 131–147 (2025)
12. Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., Finn, C.: Efficiently identifying task groupings for multi-task learning. In: Advances in Neural Information Processing Systems. vol. 34, pp. 27503–27516 (2021)
13. Guo, M., Haque, A., Huang, D.A., Yeung, S., Fei-Fei, L.: Dynamic task prioritization for multitask learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
14. Hao, J., Shen, T., Zhu, X., Liu, Y., Behera, A., Zhang, D., Chen, B., Liu, J., Zhang, J., Zhao, Y.: Retinal structure detection in octa image via voting-based multitask learning. IEEE Transactions on Medical Imaging **41**(12), 3969–3980 (2022)
15. Kim, S., Purdie, T.G., McIntosh, C.: Cross-task attention network: Improving multi-task learning for medical imaging applications. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2023 Workshops. pp. 119–128 (2023)
16. Kingma, D.P.: Adam: A method for stochastic optimization. The third International Conference on Learning Representations (2015)
17. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (1951)
18. Lin, B., Ye, F., Zhang, Y., Tsang, I.W.: Reasonable effectiveness of random weighting: A litmus test for multi-task learning. Transactions on Machine Learning Research pp. 2835–8856 (2022)
19. Liu, B., Feng, Y., Stone, P., Liu, Q.: Famo: Fast adaptive multitask optimization. In: Advances in Neural Information Processing Systems. vol. 36, pp. 57226–57243 (2023)
20. 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. 18878–18890 (2021)
21. Liu, L., Li, Y., Kuang, Z., Xue, J.H., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Towards impartial multi-task learning. In: International Conference on Learning Representations (2021)
22. 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 (CVPR). pp. 1871–1880 (2019)
23. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3730–3738 (2015)
24. Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., Fetaya, E.: Multi-task learning as a bargaining game. In: Proceedings of the 39th International Conference on Machine Learning. vol. 162, pp. 16428–16446 (2022)

25. Ong, J., Herremans, D.: Constructing time-series momentum portfolios with deep multi-task learning. Expert Systems with Applications **230**, 120587 (2023)
26. Ruder, S.: An overview of multi-task learning in deep neural networks (2017), http://arxiv.org/abs/1706.05098
27. Senushkin, D., Patakin, N., Kuznetsov, A., Konushin, A.: Independent component alignment for multi-task learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20083–20093 (2023)
28. Shen, J., Zhen, X., Worring, M., Shao, L.: Variational multi-task learning with gumbel-softmax priors. In: Advances in Neural Information Processing Systems. vol. 34, pp. 21031–21042 (2021)
29. SHI, G., Li, Q., Zhang, W., Chen, J., Wu, X.M.: Recon: Reducing conflicting gradients from the root for multi-task learning. In: The Eleventh International Conference on Learning Representations (2023)
30. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Proceedings of the 12th European Conference on Computer Vision (ECCV). pp. 746–760 (2012)
31. Sodhani, S., Zhang, A., Pineau, J.: Multi-task reinforcement learning with context-based representations. In: Proceedings of the 38th International Conference on Machine Learning. vol. 139, pp. 9767–9779 (2021)
32. Standley, T., Zamir, A., 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. vol. 119, pp. 9120–9132 (2020)
33. Tian, Y., Bai, K.: End-to-end multitask learning with vision transformer. IEEE Transactions on Neural Networks and Learning Systems **35**(7), 9579–9590 (2024)
34. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355 (2018)
35. Wulfmeier, M., Abdolmaleki, A., Hafner, R., Springenberg, J.T., Neunert, M., Siegel, N., Hertweck, T., Lampe, T., Heess, N., Riedmiller, M.: Compositional transfer in hierarchical reinforcement learning. In: Proceedings of Robotics: Science and Systems (2020)
36. Yang, R., Xu, H., WU, Y., Wang, X.: Multi-task reinforcement learning with soft modularization. In: Advances in Neural Information Processing Systems. vol. 33, pp. 4767–4777 (2020)
37. 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. 5824–5836 (2020)
38. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Proceedings of the Conference on Robot Learning (2020)
39. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
40. Zhuang, J., Tang, T., Ding, Y., Tatikonda, S.C., Dvornek, N., Papademetris, X., Duncan, J.: Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In: Advances in Neural Information Processing Systems. vol. 33, pp. 18795–18806 (2020)