SAM with sampling for Multi-Task Optimization with global-local information

Gokul Ram Subramani*
Master of Science
Computer Science and Engineering
University at Buffalo
Buffalo, USA
gsubrama@buffalo.edu

Hao Ban* (Co-Advisor)
Doctor of Philosophy
Computer Science and Engineering
University at Buffalo
Buffalo, USA
haoban@buffalo.edu

Kaiyi Ji* (Advisor)
Assistant Professor
Computer Science and Engineering
University at Buffalo
Buffalo, USA
kaiyiji@buffalo.edu

***Abstract*—Multi-Task Learning (MTL) is a Machine Learning approach where a single model is trained to tackle multiple tasks at the same time. This could be helpful in improving the overall performance and generalization for the given application. However MTL models come with a major disadvantage of task conflicts in which improvement of model in one particular task might affect the model's performance in other tasks. This is generally caused due to gradient conflicts between several tasks. Sharpness-Aware minimization (SAM) minimizes the task loss while simultaneously reducing the sharpness of the loss landscape, Our empirical observations shows that SAM effectively mitigates task conflicts in MTL. Inspired by this observation we researched about integrating SAM into MTL with sampling. Both average loss gradient and individual task gradients as perturbations help achieve good result in MTL but combining them both remains unclear. Also not all task specific gradients are useful perturbations for MTL since most useful informations might be only within certain layers of the backbone, not only that effectively computing task specific gradient without additional overhead is another challenge.**

I. Introduction

Multi-task learning (MTL) is a machine learning approach where a single model tries to perform multiple tasks simulataneously by leveraging shared informations between tasks[1]. This improves data efficiency and enhances generalization across all tasks[2]. This approach is widely used in several applications, including natural language processing[3; 4; 5], computer vision [6; 7], speech recognition [8; 9] etc.

One major challenge that MTL encounters is the problem of task conflicts where improvement of model in one particular task might lead to degradation of performance for other tasks. This is majorly caused due to gradient conflicts that have varying magnitudes and directions for different tasks [10].

Sharpness-Aware Minimization (SAM) [11] focuses on updating the model parameter in such a way that it not only minimizes the task loss but also reduces the sharpness of the loss landscape [12], It has been found useful in application generalization [13; 14], and transfer learning [15]. But still its proper usage for Multi-task learning still remains unexplored except for the recent study [16]. The authors of this paper proposed a method called F-MTL that integrates SAM into MTL

[1]This report is submitted in fulfillment of the MS Research Project.

A. Motivation

Even though F-MTL improves performance for MTL modes, it has two big issues.

i) First is that the computational cost involved is significantly larger since applying SAM individually to each task incurs additional gradient computation and also the separate manipulation of two gradient components doubles memory and time cost, where K is the number of tasks.

ii) Second the perturbations of this method only makes use of task-specific information and neglects the **shared-information** across the tasks.
To address this issues we propose a algorithm that can efficiently apply SAM into MTL.

II. Related-works

Parameter sharing in MTL frameworks are usually done in two ways. The first method is **hard parameter** sharing in which there is a backbone layer that captures the shared representation and on top of this we have multiple task specific layers that are learned independently for each task [17; 18; 19; 20]. The second method is **soft parameter** sharing where each task has its own model and for each task we penalize the distance between different task specific model's parameters [21; 22]. In computer vision multi-task learning is used to exploit the shared features and use them for different vision tasks. The MT3DNet [23] architecture leverages Multi-task learning to perform various tasks (segmentation, monocular depth estimation and object detection) concurrently. All these tasks are integrated for 3d surgical scene reconstruction. Computational cost and efficiency can be improved by MTL [24; 25] since it allows simultaneous prediction of different tasks rather than training separate models for each task and this can be beneficial in satellite imagery masking for estimating Suspended Sediment Concentration [2].

Multi-objective Optimization (MOO) as discussed by [10] helps in MTL scenarios. In MTL it is necessary to ensure conflicts between tasks are mitigated for optimal performance, this is where MOO comes in handy for mitigating conflicts by optimizing every task's objectives simultaneously.

Loss Balancing Methods: Loss balancing method dynamically updates the objective weights using some measures in loss such as how fast the loss decreases, homeostatic uncertainty of loss, loss scale and validation loss. Dynamic Weight Average (DWA) gets the objective weight to be the ratio of training losses from last two iterations for the corresponding objective. Impartial Multi-Task learning (IMTL) aims to balance losses across each task by transforming every objective to similar loss scale. Multi-Objective Meta Learning (MOML) uses Meta Learning to adaptively tune the objective weights. This approach is formulated as a multi-objective bi-level optimization problem and the time and memory cost grows significantly as the dimension of parameter increases. Auto- has similar formulation as MOML but the multi-objective upper-level subproblem is replaced with a single-objective problem and also approximates the complex hypergradient, this make the method much efficient than MOML.

Gradient Balancing Methods: Gradient balancing method adaptively aggregates the gradient of all objectives at each iteration to find the update direction. In gradient weighting method, update direction is computed as a weighted sum of gradient of all objectives. Multiple Gradient Descent Algorithm (MGDA) aims to find direction so as to maximize the minimal decrease across objectives. Conflict-Averse Gradient Descent (CAGrad) improves the previous MGDA by making sure that the aggregated direction stays close to average gradient. In gradient manipulation, the gradient are converted to overcome conflicting gradient. Projecting Conflicting Gradients (PCGrad) is a method by which the gradients that are conflicting are projected onto the normal plane of the other objectives' gradient. By this way the gradient conflicts are reduced. Gradient Vaccine (GradVac) extends PCGrad to more generalized form, based on the cosine similarity between gradients of objectives' the corrected gradient is obtained.

III. Preliminaries

A. Multi-task learning

In MTL we try to optimize multiple objective functions, for K multiple tasks the optimization is given by

$$\min_{\theta \in \mathbb{R}^m} L = (l_1(\theta), l_2(\theta), \cdots, l_K(\theta)),$$

$\{l_i\}_{i=1}^K$ are the objectives parameterized by $\theta \in \mathbb{R}^m$. A solution is Pareto optimal if no other solution outperforms it on every objective simultaneously. In contrast, a solution is Pareto stationary when there exists a convex combination of the gradients of all objectives that sums to zero, in other words the gradients are linearly dependent. Since Pareto stationarity is a prerequisite for Pareto optimality, most multi-task learning optimizers focus on locating Pareto-stationary points.

B. Sharpness-aware minimization

Consider a model parameterized by $\theta \in \mathbb{R}^m$ and the corresponding loss function given by $l(\theta)$, we consider a small perturbation given by ϵ added to model parameters, where $\|\epsilon\| \leq \rho$. The change in loss is given by $l(\theta + \epsilon) - l(\theta)$ which indicates the sharpness of the loss landscape at θ in the direction of perturbation [11].

$$\min_{\theta \in \mathbb{R}^m} \max_{\|\epsilon\| \leq \rho} l(\theta + \epsilon)$$

The inner maximization problem $\max_{\|\epsilon\| \leq \rho} l(\theta + \epsilon)$ can be approximated using Taylor's series. This leads to the approximate perturbation as

$$\hat{\epsilon}(\theta) = \rho \nabla l(\theta) / \|\nabla l(\theta)\|,$$

This shows that the perturbation is oriented in the direction of current gradient with a small step forward. The gradient of the outer minimization is then given by

$$\begin{aligned} \nabla_\theta l(\theta + \hat{\epsilon}(\theta)) &= \frac{d(\theta + \hat{\epsilon}(\theta))}{d\theta} \cdot \nabla_\theta l(\theta)\big|_{\theta + \hat{\epsilon}(\theta)} \\ &\approx \nabla_\theta l(\theta)\big|_{\theta + \hat{\epsilon}(\theta)}, \end{aligned}$$

The approximation drops the second-order gradients to improve the computational efficiency.

IV. How SAM mitigates Task conflicts

A. Toy Example

In MTL, task conflicts arise when different tasks have gradients pointing in conflicting directions, leading to instability or degraded performance for one or more tasks. Traditional approaches try to manipulate the gradient directions directly (like PCGrad, CAGrad). In contrast, SAM alters the loss geometry to find flatter regions that are generally less sensitive to parameter perturbations and more compatible across tasks. To evaluate SAM's effectiveness in mitigating task conflicts, we conduct experiments using a toy example involving two conflicting objective functions defined as:

$$\begin{aligned} f_1(\mathbf{x}) &= (x_1 - 2)^2 + 0.5(x_2 + 1)^2 \\ f_2(\mathbf{x}) &= 0.5(x_1 + 2)^2 + (x_2 - 1)^2 \end{aligned}$$

where the parameter vector is constrained to $x_1 \in [-5, 5]$, and $x_2 \in [-3, 3]$. These two objectives are designed such that their respective minima lie in opposite quadrants of the search space, creating gradient conflicts in the shared parameter space. Unlike dynamic-weighting MTL methods like MGDA or PCGrad that attempt to directly modify gradients at each iteration, we adopt Linear Scalarization (LS), which combines task losses using static weights. To this baseline, we apply SAM to the scalarized loss and evaluate the difference in optimization trajectory. Both LS and LS with SAM are initialized from the same starting point $x_0 = (0, 0)$, but they converge to different solutions. While LS typically converges to a sharp local minimum biased towards one objective, LS with SAM leads to a flatter solution region that balances both objectives more effectively. In these flatter regions, moving along the loss surface yields minimal degradation to either objective, indicating reduced **task conflict**. Our experiments show that SAM consistently pushes the solution toward Pareto-efficient points where gradients from both tasks are better aligned. This demonstrates that SAM, by smoothing the shared loss landscape, naturally mitigates task conflicts without explicitly modifying the gradients for each task.

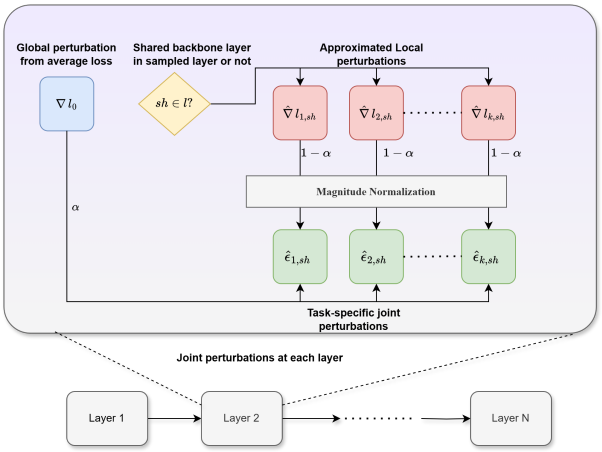

Fig. 1. SAM with sampling and global-local perturbation along with magnitude normalization to match the magnitude of global perturbation

Algorithm 1 SAM with global and sampled local perturbations

Input: Model parameters θ_0, loss functions $l_1, \cdots, l_K$, gradient manipulation MTL method $\mathcal{M}$, learning rate η, perturbation step size ρ, iteration steps T, set of sampled layers l.
Output: MTL model trained with efficient SAM
for $t = 0$ to $T-1$ **do**
  Compute average gradient $\nabla_\theta l_0(\theta_t)$
  for task $i = 1$ to K **do**
    if $sh \in l$ **then**
      Compute layerwise gradient $\hat{\nabla}_\theta l_i(\theta_{t,sh})$
    end if
    Compute perturbation $\hat{\epsilon}_{i,sh}$
    Compute gradient $g_{t,i}^{SAM}$
  end for
  Compute $d_t = \mathcal{M}(g_{t,1}^{SAM}, \cdots, g_{t,K}^{SAM})$
  Update the parameters $\theta_{t+1} = \theta_t - \eta d_t$
end for

B. Both Local and Global Information helps

We analyze the impact of incorporating two types of sharpness information on multi-task learning (MTL) performance. The first variant, referred to as **global information**, computes the gradient using the average loss across all tasks and applies a single shared perturbation to the model parameters. In contrast, the second variant, termed **local information**, calculates gradients independently for each task, leading to task-specific perturbations. Our experimental results indicate that both global and local sharpness-aware perturbations significantly improve the performance of baseline MTL methods.

V. Proposed Algorithm

From the experiments reported in Section IV we propose a efficient algorithm that makes use of both global and sampled local information as shown in Algorithm 1, which combines the benefits of G-SAM and L-SAM along with sampling to mitigate task conflicts while keeping the computation cost manageable.

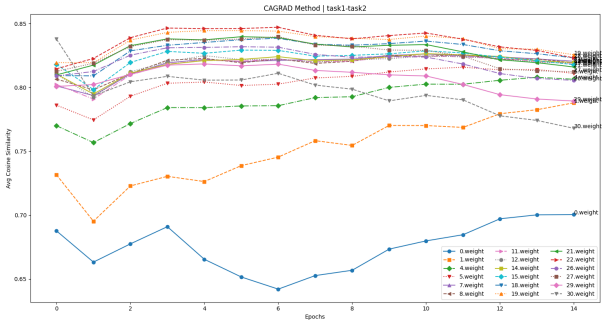

Fig. 2. Cosine Similarity of different backbone layers between 2 tasks on CelebA dataset using CAGRAD method

Compared to the traditional MTL methods, the total computational cost of the proposed algorithm involves $K+1$ gradient computations along with additional forward pass computation, making it more efficient than F-MTL [16]. On top of this task-specific gradients are approximated and only computed to those crucial layers. The formulation of the perturbation for SAM is given by:

$$\hat{\epsilon}_{i,sh} = \begin{cases} \rho \frac{\alpha \nabla_\theta l_0(\theta) + (1-\alpha)\nabla_\theta l_i(\theta_{sh})}{\|\alpha \nabla_\theta l_0(\theta) + (1-\alpha)\nabla_\theta l_i(\theta_{sh})\|} & \text{if } sh \in l \\ \rho \frac{\alpha \nabla_\theta l_0(\theta)}{\|\alpha \nabla_\theta l_0(\theta)\|} & \text{otherwise.} \end{cases}$$

l here refers to set of sampled backbone layers that are considered to be crucial layers for model learning between tasks. $\alpha \in [0, 1]$ is a tunable weight scalar which is used to control the amount of global and local information in the perturbation. For equal amount of information $\alpha = 0.5$ is chosen.

The **cosine similarity** in Figure 2,3 shows the dynamics of different backbone layers between several tasks. This can help identify the crucial layers that are shared between several tasks, and only the information from these layers in perturbation will have a major impact. Only certain layers—specifically the **first, some middle layers, and the last layers**—showed more changes in cosine similarity compared to other layers in the backbone. Let C_f, C_b represent the computation cost of a forward pass and a backward pass, the proposed algorithm has a total computational cost of $\mathbf{C_b} + 2K\mathbf{C_f}$, One backward pass to compute the average gradient and two forward passes to compute the approximated local perturbation for each task. This is much efficient than F-MTL [16] which has a total overhead of $\mathbf{KC_b} + C_{gm}$, where C_{gm} represents the cost for gradient manipulation MTL method.

VI. Conclusion

We first researched about how integrating SAM into MTL can mitigate task conflicts with the help of both average gradients and task specific gradients. We then proposed a efficient algorithm that combines both global and sampled local information that helps in training MTL models with better generalization performance and reduced computational cost. We will also explore effective and efficient methods in

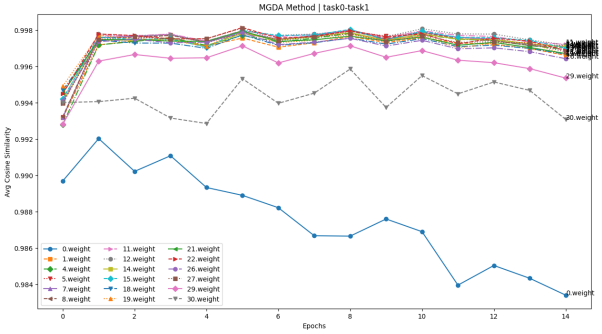

Fig. 3. Cosine Similarity of different backbone layers between 2 tasks on CelebA dataset using MGDA method

future for mitigating general task conflicts in MTL using Low-Rank approximations.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor, Professor Kaiyi Ji, whose expert guidance and encouragement made this work in Multi-task optimization possible. I am also deeply thankful to Hao Ban for his unwavering support and his guidance throughout this research project without which this work wouldn't have become possible. Finally, I extend my appreciation to all members of the Optimization for Machine Learning and Networks Lab for their encouragement and assistance.

REFERENCES

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

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

[3] S. Chen, Y. Zhang, and Q. Yang, "Multi-task learning in natural language processing: An overview," *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–32, 2024.

[4] J. Yu, Y. Dai, X. Liu, J. Huang, Y. Shen, K. Zhang, R. Zhou, E. Adhikarla, W. Ye, Y. Liu *et al.*, "Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras," *arXiv preprint arXiv:2404.18961*, 2024.

[5] Z. Zhang, W. Yu, M. Yu, Z. Guo, and M. Jiang, "A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 943–956.

[6] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[7] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[8] Y. Tseng, L. Berry, Y.-T. Chen, I.-H. Chiu, H.-H. Lin, M. Liu, P. Peng, Y.-J. Shih, H.-Y. Wang, H. Wu *et al.*, "Av-superb: A multi-task evaluation benchmark for audio-visual representation models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 6890–6894.

[9] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Conditional language models for speech recognition, synthesis, and translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[10] W. Chen, X. Zhang, B. Lin, X. Lin, H. Zhao, Q. Zhang, and J. T. Kwok, "Gradient-based multi-objective deep learning: Algorithms, theories, applications, and beyond," *arXiv preprint arXiv:2501.10945*, 2025.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*.

[12] M. Abbas, Q. Xiao, L. Chen, P.-Y. Chen, and T. Chen, "Sharp-maml: Sharpness-aware model-agnostic meta learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 10–32.

[13] P. Wang, Z. Zhang, Z. Lei, and L. Zhang, "Sharpness-aware gradient matching for domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3769–3778.

[14] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park, "Swad: Domain generalization by seeking flat minima," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 405–22 418, 2021.

[15] D. Bahri, H. Mobahi, and Y. Tay, "Sharpness-aware minimization improves language model generalization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 7360–7371.

[16] H. Phan, L. Tran, N. N. Tran, N. Ho, D. Phung, and T. Le, "Improving multi-task learning via seeking task-based flat regions," *arXiv preprint arXiv:2211.13723*, 2022.

[17] F. Heuer, S. Mantowsky, S. Bukhari, and G. Schneider, "Multitask-centernet (mcn): Efficient and diverse multitask learning using an anchor free approach," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 997–1005.

[18] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1439–1449.

[19] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504*, 2019.

[20] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision us-

ing diverse datasets and limited memory," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6129–6138.

[21] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4822–4829.

[22] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3205–3214.

[23] M. Parab, P. Lendave, J. Kim, T. Q. D. Nguyen, and P. Ingle, "Mt3dnet: Multi-task learning network for 3d surgical scene reconstruction," *arXiv preprint arXiv:2412.03928*, 2024.

[24] R. Daroya, L. V. Lucchese, T. Simmons, P. Prum, T. Pavelsky, J. Gardner, C. J. Gleason, and S. Maji, "Improving satellite imagery masking using multi-task and transfer learning," *arXiv preprint arXiv:2412.08545*, 2024.

[25] Z. Qi, J. Chen, S. Wang, B. Liu, H. Zheng, and C. Wang, "Optimizing multi-task learning for enhanced performance in large language models," *arXiv preprint arXiv:2412.06249*, 2024.