Delve into the Applicability of Advanced Optimizers for Multi-Task Learning

Zhipeng Zhou[1,4,†], **Linxiao Cao**[2,†], **Yiming Cao**[1], **Pengcheng Wu**[1], **Peilin Zhao**[3], **Chunyan Miao**[1,⋆]

[1]Nanyang Technological University
[2]Hong Kong University of Science and Technology (Guangzhou)
[3]School of Artificial Intelligence, Shanghai Jiao Tong University
[4]Jinan-NTU Green Technology Research Institute
zzpustcml@gmail.com, lcao950@connect.hkust-gz.edu.cn, peilinzhao@sjtu.edu.cn
{yiming.cao, pengchengwu, ascymiao}@ntu.edu.sg

Abstract

Multi-Task Learning (MTL) is a fundamental problem in machine learning that has been extensively studied over the past decade. Recently, a variety of optimization-based MTL approaches have been proposed to jointly learn multiple tasks by modifying the optimization trajectory. In this paper, we argue that the design of mainstream optimization-based MTL methods implicitly assumes compatibility with momentum-free optimizers, which may limit the understanding of their effectiveness in modern training regimes. In practice, the instantaneously derived gradients from MTL operations contribute only marginally to the final parameter updates when momentum is present, resulting in an amortized rather than instant de-conflicting effect. Moreover, this amortized behavior can be further degraded when gradient directions change rapidly. To address this issue and enable the effective integration of mainstream MTL methods with advanced optimizers (e.g., Adam and Muon), we propose `APT` (Applicability of advanced oPTimizers), a lightweight framework featuring an adaptive momentum mechanism that balances the trade-off between instant and amortized de-conflicting. Furthermore, we give a conditional interpretation of the Muon optimizer as an implicit MTL learner, and we introduce a lightweight direction preservation strategy to better align with its orthogonalization process. Extensive experiments across four standard MTL benchmarks demonstrate that `APT` enhances most evaluated method–dataset combinations, yielding substantial performance gains in several settings. Code is available at `https://github.com/zzpustc/APT`.

1 Introduction

Traditional machine learning typically relies on customized models tailored with task-specific knowledge. However, in modern real-world scenarios, the requirement to handle multiple tasks simultaneously has made the development of Multi-Task Learning (MTL) models increasingly essential.

MTL is a mature field with diverse methodologies, including architecture-based approaches [Zhao et al., 2024, Heuer et al., 2021], optimization-based methods [Liu et al., 2021a, 2023], and task relationship modeling [Song et al., 2022, Fifty et al., 2021]. This paper focuses exclusively on optimization-based MTL, which assumes a fixed architecture comprising a shared backbone and task-specific branches [Liu et al., 2019b]. These methods aim to design effective re-weighting strategies for task gradients or losses.

† Equal contribution. ⋆ Corresponding author.

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

Recent benchmarks [Ban and Ji, 2024, Zhou et al., 2025b] show that advanced optimization-based MTL methods achieve competitive performance, sometimes even surpassing Single-Task Learning (STL) reference models. Despite this strong empirical performance, their effectiveness is not yet fully understood under modern training regimes. Most existing approaches [Ban and Ji, 2024, Liu et al., 2021a, Sener and Koltun, 2018] rely on intuitive, step-wise Pareto-improving operations to approach the Pareto front. However, their analyses of de-conflicting and convergence often focus on the instantaneous aggregate, without accounting for the subsequent momentum transformation, which is inconsistent with both current training practices and their actual implementations. In practice, **almost all** optimization-based MTL methods adopt Adam optimizers (with momentum) across standard benchmarks. As a result, the expected step-wise de-conflicting (**instant de-conflicting**) may not hold.

As illustrated in Figure 1, even when a de-conflicted direction g_{mtl} is obtained, the exponential moving average (EMA) in modern optimizers allows the momentum term g_{mom} to dominate, producing an update direction g_{update} that may still contain conflicts. Therefore, instead of achieving instant de-conflicting as usually expected, we show that these methods perform **amortized de-conflicting**. While this mechanism remains effective under stable optimization dynamics, it may fail when gradient directions change rapidly. To address this issue, we propose an adaptive momentum mechanism that balances instant and amortized de-conflicting, improving the compatibility of optimization-based MTL with modern optimizers. We focus on Adam, AdamW, and Muon as representative optimizers in our study.

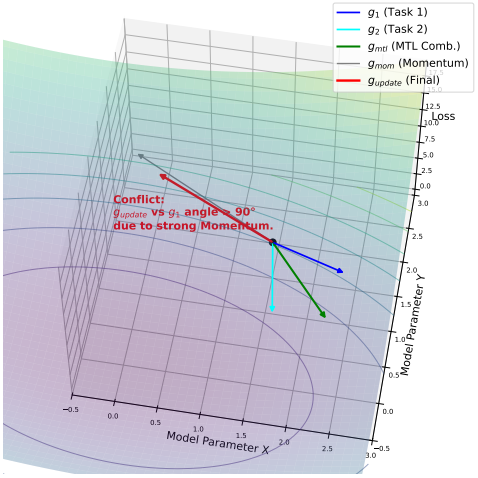


Figure 1: Illustration of the impact of EMA in advanced optimizer for MTL. Here we adopt $g_{update} = 0.9 * g_{mom} + 0.1 * g_{mtl}$ as the example.

In addition, as a secondary contribution, we show that, under explicit task–spectrum loading conditions, the Muon optimizer can be interpreted as an implicit MTL learner. Based on this observation, we further propose a lightweight direction preservation (LDP) method to enhance its performance. In summary, our contributions are as follows:

- We show, both theoretically and empirically, that mainstream optimization-based MTL methods primarily achieve amortized rather than instant de-conflicting. This observation, which has been largely overlooked in prior work, indicates their limitations under rapidly changing gradient geometry.
- We propose `APT` (Applicability of advanced oPTimizers), a lightweight framework with an adaptive momentum mechanism that balances instant and amortized de-conflicting, along with auxiliary tracking and conditional stationarity bounds.
- We give a conditional view of the Muon optimizer as an implicit MTL learner and propose an LDP method to improve its input gradient.
- Extensive experiments on standard MTL benchmarks show that our method improves most evaluated combinations and achieves competitive performance.

2 Related Work

2.1 Optimization-based Multi-Task Learning

In optimization-based MTL, a wide range of methods have been proposed [Zhou et al., 2025a,c, Cao et al., 2026], which can be broadly categorized into gradient-based and loss-based approaches. Gradient-based methods operate directly on task gradients with respect to shared parameters and aim to resolve conflicts through gradient manipulation. For example, MGDA [Sener and Koltun, 2018] seeks a convex combination of task gradients with minimum norm. PCGrad [Yu et al., 2020] addresses gradient conflicts by projecting conflicting gradients onto orthogonal subspaces. CAGrad [Liu et al., 2021a] balances minimum-norm optimization with convergence guarantees. Align-MTL [Senushkin

et al., 2023] performs singular value decomposition (SVD) on task gradients and re-weights gradient components in the resulting subspace. Nash-MTL [Navon et al., 2022] and FairGrad [Ban and Ji, 2024] both formulate MTL as a fair allocation problem across tasks, with Nash-MTL being a special case under certain settings. COST [Zhou et al., 2025b] tackles gradient conflicts by seeking alternative update directions that lie on the same loss level set.

In contrast, loss-based methods adjust task weights based solely on task losses computed from model outputs, without explicitly using task gradients. As a result, they are typically more lightweight but often underperform gradient-based approaches due to the lack of gradient-level information. Early loss-based methods include RLW [Lin et al., 2021], which samples task weights from a random distribution, and Uncertainty Weighting [Kendall et al., 2018], which derives weights based on task-specific predictive uncertainty. More recent loss-based approaches are commonly formulated within a bi-level optimization framework, differing mainly in their lower-level task-weight update strategies. For instance, FAMO [Liu et al., 2023] amortizes the MGDA objective over time using a first-order approximation. Go4Align [Shen et al., 2024] introduces dynamic task grouping in the lower-level optimization. LDC-MTL [Xiao et al., 2025] employs a routing network to regulate loss discrepancies during training.

2.2 Advanced Optimizers

Advanced optimizers in deep learning are dominated by the Adam family, which builds on exponential moving averages of first- and second-order moments to provide per-parameter adaptive steps [Kingma and Ba, 2014]. Subsequent variants target stability, generalization, and theoretical issues: AdaMax replaces the second-moment normalization with an ℓ_∞-based scheme; AMSGrad [Reddi et al., 2019] introduces a non-decreasing second-moment accumulator to address non-convergence behaviors in Adam-type methods; AdamW decouples weight decay from the adaptive update, improving regularization and often empirical generalization [Loshchilov and Hutter, 2017]; RAdam [Liu et al., 2019a] rectifies the early-stage variance of the adaptive learning rate, reducing sensitivity to warmup heuristics; and AdaBound [Luo et al., 2019] dynamically bounds learning rates to interpolate between adaptive methods and SGD. More recently, Muon [Jordan et al., 2024] departs from element-wise adaptivity by applying matrix orthogonalization to parameter updates (notably for hidden-layer weight matrices), aiming to improve conditioning and achieve higher compute-efficiency in large-scale pretraining.

Discussion with Counterparts. To the best of our knowledge, UPGrad [Quinton and Rey, 2024] is the most related work with this paper. In its implementation, it adopts both SGD and Adam optimizers to demonstrate the compatibility of the proposed method [1], but does not study the temporal adaptation considered here. Adaptive momentum is also studied outside this setting: DEMON [Chen et al., 2022] uses a training-progress schedule, AMM [Topollai and Choromanska, 2026] adapts momentum from gradient and memory geometry, and YellowFin [Zhang and Mitliagkas, 2019] jointly tunes learning rate and momentum. MoCoGrad [Chai et al., 2024] instead calibrates task-wise momentum using task conflicts. Our rule uses consecutive MTL aggregates, and we compare these alternatives in Section 5.2.

3 Motivation

3.1 Preliminary

Advanced optimizers typically share a common momentum mechanism; here, we use Adam as a representative example to illustrate this process. The momentum update is defined as:

$$m_t = \beta * m_{t-1} + (1 - \beta) * G_t \tag{1}$$

where m_t represents the momentum (the first moment vector) at time step t. β is a hyperparameter, typically set to 0.9 or higher. G_t is the gradient derived from the current step. In the context of MTL, G_t is the aggregated gradient produced by a specific MTL algorithm (e.g., MGDA, CAGrad, or Nash-MTL, etc). For detailed introduction of the AdamW and Muon optimizers, please refer to the Appendix Sec. C. In the following, we first reveal a fundamental mismatch between the momentum mechanism used in advanced optimizers and the objectives of MTL, uncovering the actual de-conflicting effect MTL achieves, thus motivate our design in the next section.

[1] We re-implement and augment UPGrad with our `APT`, please refer to Appendix Sec. B.4 for more results.

Table 1: Analysis on β across various MTL approaches and advanced optimizers on *CityScapes*. AdamW is set with weight decay as 1.e-4 to distinguish with Adam. All results are average on 3 random seeds. The best scores are provided in gray. More results please refer to Table 9.

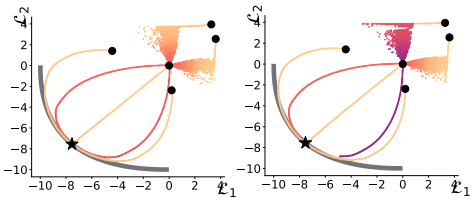
β	MGDA			CAGrad		
	Adam	AdamW	Muon	Adam	AdamW	Muon
default	7.29 ± 4.61	6.41 ± 5.71	3.37 ± 4.02	12.38 ± 1.49	10.57 ± 1.80	-2.37 ± 0.54
0.6	9.21 ± 4.12	5.82 ± 3.37	4.14 ± 1.10	9.78 ± 1.14	10.88 ± 1.54	1.95 ± 0.90
0.2	9.25 ± 3.75	5.17 ± 3.24	19.39 ± 6.64	10.78 ± 3.03	10.07 ± 0.96	81.12 ± 9.13
0.0	3.52 ± 1.50	10.23 ± 4.68	33.35 ± 6.61	9.11 ± 0.97	10.08 ± 2.58	120.78 ± 36.70
β	Nash-MTL			FairGrad		
	Adam	AdamW	Muon	Adam	AdamW	Muon
default	24.87 ± 1.57	23.42 ± 1.61	-4.22 ± 0.66	4.23 ± 3.20	4.04 ± 0.66	6.04 ± 1.16
0.6	22.59 ± 1.39	21.62 ± 1.89	3.15 ± 2.97	4.81 ± 2.77	3.53 ± 4.19	1.63 ± 3.05
0.2	20.45 ± 1.68	19.05 ± 2.30	77.04 ± 27.68	2.86 ± 1.13	2.94 ± 0.88	44.29 ± 25.49
0.0	20.68 ± 2.04	19.68 ± 0.99	92.93 ± 5.26	2.38 ± 0.90	3.26 ± 1.22	55.87 ± 28.48

3.2 Does MTL Work Like We Expect?

Most prior work on MTL implicitly assumes a momentum-free optimization regime. However, modern optimizers rely heavily on EMA momentum. This raises a fundamental question:

Does MTL retain its desired properties under momentum?

Momentum Undermines Instant De-conflicting. We empirically study the interaction between momentum and gradient de-conflicting on CityScapes using FairGrad. Specifically, we measure the cosine similarity between the combined gradient g_{com} and each task gradient $\{g_i\}_{i=1}^{K}$. Results are shown in Figure 3(a)–(c). For LS, g_{com} is the mean of task gradients. For FairGrad, g_{com} is the conflict-aware aggregated gradient. Under momentum, g_{com} corresponds to the EMA update defined in Eqn. 1, where the instantaneous gradient G_t is produced by FairGrad.

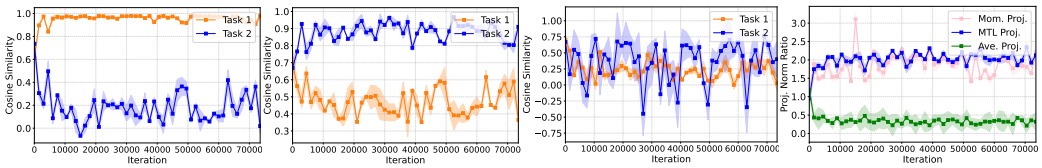


(a) $\beta_1 = 0.9$. (b) $\beta_1 = 0$.

Figure 2: Effect of momentum on toy example.

As shown in Figure 3, LS produces highly imbalanced and often conflicting gradients. FairGrad addresses this issue by constructing a descent direction that is non-conflicting across tasks. However, this property is not preserved under momentum. The EMA introduces a bias toward historical gradients, yielding an update direction that is less aligned with the current task gradients. Consequently, a non-conflicting aggregate need not yield a non-conflicting first moment; not every MTL method guarantees common descent in the first place. This effect is further supported by the results in Table 1. Across MGDA, CAGrad, Nash-MTL, and FairGrad, decreasing β often improves Adam performance, suggesting that placing greater—though not exclusive—weight on the current MTL gradient G_t is beneficial. However, the optimal value of β varies across benchmarks (see Table 9 for results on NYUv2), motivating adaptation to changing gradient directions, as discussed next. The effect is not monotone, and very low momentum can substantially degrade Muon.

Why Current MTL Still Works. Despite this limitation, momentum remains essential for stable MTL optimization. The EMA mechanism reduces gradient noise and smooths optimization trajectories. Following [Liu et al., 2023], Figure 2 compares trajectories with $\beta_1 = 0.9$ and $\beta_1 = 0$. Without momentum, optimization becomes highly oscillatory and may fail to reach the Pareto front. This highlights a fundamental trade-off: momentum improves stability but distorts instant de-conflicting.



(a) Similarity over Ave. (LS). (b) Similarity over FairGrad. (c) Similarity over Momentum.(d) Project Norm Comparison.

Figure 3: Similarity and project norm comparison.

On the other hand, consider the expected task gradient $\bar{g}_i$ and the stochastic gradient $\hat{g}_{i,t} = \bar{g}_i + \epsilon_{i,t}$, where $\epsilon_{i,t}$ is zero-mean noise. For a locally fixed $\bar{g}_i$ and $m_0 = 0$, the momentum update satisfies:

$$\langle \bar{g}_i, m_t \rangle = \underbrace{(1-\beta)\sum_{k=1}^{t} \beta^{t-k} \langle \hat{g}_{i,k}, g_{mtl,k} \rangle}_{\text{Term A (Step-wise Alignment)}} - \underbrace{(1-\beta)\sum_{k=1}^{t} \beta^{t-k} \langle \epsilon_{i,k}, g_{mtl,k} \rangle}_{\text{Term B (Noise Correction)}} \tag{2}$$

Term A is nonnegative when each aggregate is non-conflicting. Zero-mean noise alone does not make Term B vanish, since the aggregate depends on the same noisy gradients. Cancellation requires an additional orthogonality condition; more generally, a positive expected Term A must dominate the expected noise correction (which is usually observed). Under these conditions, momentum preserves a *long-term* alignment with each task gradient. We refer to this as *amortized de-conflicting*. The proof of Eqn. (2) is provided in Appendix Sec. E.

Empirically, Figures 3(a)–(c) show that momentum maintains balanced similarity across tasks, preventing dominance by any single task. Additionally, we compute the projection norms of the momentum-based direction onto each task gradient and report their projection norm ratios to enable a more rigorous comparison. The results are shown in Figure 3(d). Note that the curves in Figure 3(d) are sampled and smoothed to highlight the overall trend. As observed, MTL methods achieve a clear re-balancing effect compared to the LS baseline. Notably, momentum-based updates exhibit a similar re-balancing behavior, although with more pronounced fluctuations.

3.3 The Flaw of Amortized De-Conflicting

The above analysis relies on a key assumption: the gradient geometry evolves smoothly over time. This assumption can fail when gradient directions change abruptly (see the empirical traces in Appendix Sec. B.1). Such changes may arise from curvature, minibatch noise, or changing task weights.

Consider a transition at step T, where the expected gradients shift from $\bar{g}_i^{(old)}$ to $\bar{g}_i^{(new)}$. In this illustrative transition model, assume that before the transition momentum is well aligned:

$$\mathbb{E}[m_{T-1}] = \bar{g}_{mtl}^{(old)}, \quad \langle \bar{g}_i^{(old)}, \bar{g}_{mtl}^{(old)} \rangle > 0.$$

After the transition, the gradient geometry changes abruptly. The stale momentum becomes adversarial:

$$\langle \bar{g}_i^{(new)}, \bar{g}_{mtl}^{(old)} \rangle = -C, \quad C > 0,$$

while the expected new MTL gradient has the same positive task projection throughout the analyzed segment:

$$\langle \bar{g}_i^{(new)}, \bar{g}_{mtl}^{(new)} \rangle = S, \quad S > 0.$$

The expected first-moment projection becomes:

$$\langle \bar{g}_i^{(new)}, \mathbb{E}[m_t] \rangle = \beta^{\Delta t+1}(-C) + (1-\beta^{\Delta t+1})S. \tag{3}$$

where $t = T + \Delta t$. To maintain non-conflicting updates, it requires:

$$\frac{1-\beta^{\Delta t+1}}{\beta^{\Delta t+1}} \geq \frac{C}{S}. \tag{4}$$

This reveals the fundamental limitation of amortized de-conflicting: the left-hand side quantifies the *momentum forgetting rate*, while the right-hand side captures the *conflict-to-signal ratio*. Please refer to Appendix Sec. E.2 for the full proof of above analysis.

Immediate Failure Regime. At $\Delta t = 0$ and $\beta = 0.9$, we have $(1-\beta)/\beta \approx 0.11$. Thus, whenever $C/S > 0.11$, the update becomes negatively aligned with the true gradient: $\langle \bar{g}_i^{(new)}, \mathbb{E}[m_t] \rangle < 0$. In this regime, each optimization step harms at least one task, leading to sustained Pareto violations. Due to slow momentum decay, this effect persists over multiple iterations.

Implication. This analysis illustrates how amortized de-conflicting can fail in non-stationary regimes. The stale-momentum calculation is generic; its MTL relevance is that agreement with an aggregate can hide disagreement with an individual task. It motivates dynamically adapting β to rapidly discard stale gradients, which leads to our proposed method `APT`.

4 Principal Design

4.1 Adaptive Momentum: A Simple Solution

In this section, we introduce a simple adaptive momentum strategy that dynamically balances instant and amortized de-conflicting based on temporal gradient consistency. Intuitively, when consecutive aggregate gradients are well aligned, a larger momentum coefficient should be employed to capitalize on temporal filtering and stability while maintaining amortized de-conflicting. Conversely, when their directions change rapidly, a lower momentum is preferred to prioritize the immediate resolution of task conflicts.

Following this intuition, we utilize the cosine similarity between gradients of adjacent steps as a directional-consistency signal:

$$\rho_t = \cos(G_t, G_{t-1}) \tag{5}$$

where G_t is derived from the MTL algorithm. A high ρ_t indicates similar aggregate directions, while a low ρ_t indicates directional change. This signal is affected by minibatch noise and task weights as well as parameter movement; it is neither a direct curvature estimate nor a measure of inter-task conflict. We then adaptively adjust the momentum coefficient β_t as follows:

$$\beta_t = \beta_{min} + (\beta_{max} - \beta_{min}) \cdot \mathrm{Clip}(\frac{\rho_t + 1}{2}, 0, 1) \tag{6}$$

where $\mathrm{Clip}(\cdot, 0, 1)$ ensures the similarity metric remains within a valid range. The hyperparameters β_{max} and β_{min} define the operational bounds of the momentum, which are fixed at $(\beta_{min}, \beta_{max}) = (0.1, 0.9)$ for Adam and $(0.5, 0.9)$ for Muon across datasets, methods, and seeds, without test-set selection.

Auxiliary Analysis. The moment updates are $m_t = \beta_t m_{t-1} + (1-\beta_t) G_t$ and $v_t = \beta_2 v_{t-1} + (1-\beta_2) G_t^2$. Incorporating the cumulative-product normalization factors $\rho_{1,t} = 1 - \prod_{j=1}^{t} \beta_j$ and $\rho_{2,t} = 1 - \beta_2^t$, the parameter update is $\theta_{t+1} = \theta_t - \eta \hat{H}_t^{-1} \hat{m}_t$, where $\hat{H}_t = \mathrm{diag}(\sqrt{\hat{v}_t} + \varepsilon)$. To facilitate tracking error analysis, we construct an equivalent update step $\theta_{t+1} = \theta_t - \tilde{\eta}_t \hat{H}_t^{-1} m_t$, where $\tilde{\eta}_t = \eta / \rho_{1,t}$ serves as an effective learning rate. Since $\beta_t \le \beta_{max} < 1$, $\tilde{\eta}_t$ is strictly bounded within $[\eta, \tilde{\eta}_{max}]$.

We define the momentum tracking error as $\delta_t = m_t - G_t$, which captures the discrepancy between historical momentum and the instantaneous MTL gradient. It satisfies

$$\delta_t = \beta_t \delta_{t-1} + \beta_t (G_{t-1} - G_t). \tag{7}$$

For $\beta_t \le \beta_{max} < 1$, the average squared tracking error is bounded by a constant times the average of $\beta_t^2 \|G_t - G_{t-1}\|^2$, up to a vanishing initial term. This motivates reducing stale momentum, although cosine ignores magnitude changes and does not guarantee a smaller bound on every trajectory. The complete tracking and surrogate-stationarity results are in Appendix F. These are auxiliary optimizer bounds, not a model of task crosstalk or a guarantee of full-sequence Pareto convergence. Moreover, first-moment alignment need not survive Adam's preconditioning; Section 5.2 therefore evaluates the executed direction d_t, defined by $\theta_{t+1} = \theta_t - \eta_t d_t$.

4.2 Muon for MTL

In this subsection, we first give conditions under which Muon can be interpreted as an MTL learner, then we propose a light direction preservation method for Muon in addition to our main framework APT.

Muon as a Conditional MTL Learner. Muon applies an orthogonalization operator via the Newton-Schulz iteration:

$$\theta_{t+1} = \theta_t - \eta O_t, \quad O_t = \text{Newton-Schulz}(G_t), \tag{8}$$

which approximates the polar factor of its input. For this idealized analysis, G_t denotes that input matrix; in practical Muon it is momentum-based rather than the raw aggregate. The conditional projection analysis in Appendix G shows how this transformation can reduce a task-projection ratio under nonnegative, ordered task loadings. Singular directions are not task identities, so this interpretation does not imply universal task balancing.

Light Direction Preservation. Although Muon can have a task-rebalancing effect under these conditions, it still has some preferences on the input aggregated gradient G_t. Since this paper only regards Muon for MTL as a secondary contribution, we put a preliminary discussion of the gradient choice for Muon in Appendix Sec. B.2.

According to the discussion, we aim to preserve the rich geometric directions of task gradients and maintain a high effective rank, as a useful input property for Muon's orthogonalization process. To retain the benefits of centered compromise—similar to the objective in CAGrad—we utilize the mean gradient $g_{mean} = \frac{1}{K}\sum_{i=1}^{K} g_i$ as an anchor direction. We then compute the task-wise cosine similarity between each individual task gradient g_i and the anchor g_{mean}:

$$c_i = \cos(g_i, g_{mean}) \tag{9}$$

To ensure the aggregated gradient captures the diverse directional information from all tasks rather than being dominated by a subset, we apply a non-linear gating mechanism:

$$\omega_i = \text{Sigmoid}(\xi \cdot c_i) \tag{10}$$

where $\text{Sigmoid}(\cdot)$ denotes the Sigmoid function and ξ is a scaling hyperparameter. The final aggregated gradient is formulated as: $G_t = \sum_{i=1}^{K} \omega_i g_i, \quad \omega_i = K \cdot \frac{\omega_i}{\sum_{i=1}^{K} \omega_i}$. This mechanism serves as a computationally efficient alternative to complex solvers like CAGrad or Nash-MTL. By bypassing the need for numerical iterations or quadratic programming (QP) solvers, it provides a soft aggregation for Muon's spectral orthogonalization (refer to Appendix Sec. B.2 for effective-rank diagnostics). This is not a guarantee of high rank or common descent. It is important to note that this approach is a lightweight framework designed to **complement Muon**, rather than a standalone MTL algorithm.

5 Performance Evaluation

Benchmarks: We evaluate our method on four widely adopted MTL benchmarks: Cityscapes [Cordts et al., 2016], NYUv2 [Silberman et al., 2012], CelebA [Liu et al., 2015], and QM9 [Ramakrishnan et al., 2014]. Following the experimental protocols established by FAMO [Liu et al., 2023], we categorize these datasets as follows:
Scene Understanding: Cityscapes contains 5,000 annotated street-view images used for two tasks: semantic segmentation and depth estimation. NYUv2 is a three-task benchmark comprising 1,449 annotated indoor images for semantic segmentation, depth estimation, and surface normal prediction.
Large-scale Classification: CelebA consists of 200,000 facial images. We treat the classification of 40 binary facial attributes as 40 distinct tasks.
Graph Regression: QM9 is a standard benchmark for Graph Neural Network (GNN) research. It includes over 130,000 stable organic molecules represented as graphs with annotated node and edge features. We evaluate this as an 11-task MTL regression problem.

Baselines: We compare it with the following baselines: Linear Scalarization (LS), Scale-Invariant (SI), Random Loss Weighting (RLW) as described in [Lin et al., 2021], Dynamic Weight Average (DWA) from [Liu et al., 2019b], Uncertainty Weighting (UW) detailed in [Kendall et al., 2018], MGDA from [Sener and Koltun, 2018], GradDrop presented in [Chen et al., 2020], PCGrad as in

Table 2: **Scene understanding** (*NYUv2*, 3 tasks). We report MTAN model performance averaged over 3 random seeds. The best scores are provided in gray, and the second scores are underlined.

Method	Segmentation ↑		Depth ↓		Surface Normal					MR ↓	Δm% ↓
					Angle Distance ↓		Within t° ↑				
	mIoU	Pix. Acc.	Abs. Err.	Rel. Err.	Mean	Median	11.25	22.5	30		
Independent	38.30	63.76	0.68	0.28	25.01	19.21	30.14	57.20	69.15	-	-
LS	39.29	65.33	0.55	0.23	28.15	23.96	22.09	47.50	61.08	13.33	5.46
RLW	37.17	63.77	0.58	0.24	28.27	24.18	22.26	47.05	60.62	16.78	7.67
DWA	39.11	65.31	0.55	0.23	27.61	23.18	24.17	50.18	62.39	12.56	3.49
Uncertainty	36.87	63.17	0.54	0.23	27.04	22.61	23.54	49.05	63.65	12.89	4.01
MGDA	30.47	59.90	0.61	0.26	24.88	19.45	29.18	56.88	69.36	10.00	1.47
GradDrop	39.39	65.12	0.55	0.23	27.48	22.96	23.38	49.44	62.87	12.89	3.61
PCGrad	38.06	64.64	0.56	0.23	27.41	22.80	23.86	49.83	63.14	13.56	3.83
CAGrad	39.79	65.49	0.55	0.23	26.31	21.58	25.61	52.36	65.58	10.11	0.29
IMTL	39.35	65.60	0.54	0.23	26.02	21.19	26.20	53.13	66.24	9.00	-0.59
Nash-MTL	40.13	65.93	0.53	0.22	25.26	20.08	28.40	55.47	68.15	5.33	-4.04
FAMO	38.88	64.90	0.55	0.22	25.06	19.57	29.21	56.61	68.98	9.22	-4.10
FairGrad	39.74	66.01	0.54	0.22	24.84	19.60	29.26	56.58	69.16	5.11	-4.66
COST	38.06	64.71	0.54	0.23	24.47	18.80	30.84	58.25	70.30	5.00	-5.39
Muon	39.00	65.36	0.51	0.23	25.27	20.25	28.57	55.11	67.90	7.89	$-3.15_{\pm 1.48}$
APT-Muon	42.47	67.92	0.49	0.22	25.22	20.19	28.54	55.12	67.95	6.00	$-5.54_{\pm 0.96}$
APT-CG	40.51	66.32	0.53	0.22	25.77	20.65	27.24	54.25	67.15	7.56	$-2.67_{\pm 0.84}$
APT-FAMO	37.66	64.57	0.54	0.23	24.87	19.11	30.12	57.50	69.58	7.67	$-4.35_{\pm 2.23}$
APT-Fair	38.97	65.07	0.54	0.22	24.44	18.85	30.84	58.12	70.22	6.11	$-6.04_{\pm 0.22}$

Table 3: **Scene understanding** (*CityScapes*, 2 tasks). MTAN means over three random seeds.

Method	Segmentation ↑		Depth ↓		MR ↓	Δm% ↓
	mIoU	Pix.	Abs.	Rel.		
Independent	74.01	93.16	0.0125	27.77	-	-
LS	75.18	93.49	0.0155	46.77	12.25	22.60
RLW	74.57	93.41	0.0158	47.79	15.50	24.37
DWA	75.24	93.52	0.0160	44.37	12.25	21.43
Uncertainty	72.02	92.85	0.0140	30.13	11.75	5.88
MGDA	68.84	91.54	0.0309	33.50	15.50	44.14
GradDrop	75.27	93.53	0.0157	47.54	11.75	23.67
PCGrad	75.13	93.48	0.0154	42.07	12.50	18.21
CAGrad	75.16	93.48	0.0141	37.60	11.25	11.58
IMTL	75.33	93.49	0.0135	38.41	9.50	11.04
Nash-MTL	75.41	93.66	0.0129	35.02	5.75	6.82
FAMO	74.54	93.29	0.0145	32.59	12.25	8.13
FairGrad	75.72	93.68	0.0134	32.25	5.25	5.18
COST	75.73	93.53	0.0133	31.53	5.25	4.30
Muon	78.65	94.49	0.0122	27.68	1.50	$-2.56_{\pm 1.04}$
APT-Muon	78.23	94.36	0.0121	27.09	1.50	$-3.09_{\pm 1.16}$
APT-CG	75.79	93.65	0.0129	38.13	6.25	$9.31_{\pm 1.21}$
APT-FAMO	75.11	93.48	0.0138	35.39	11.25	$9.03_{\pm 1.23}$
APT-Fair	74.29	93.25	0.0125	30.36	9.75	$2.22_{\pm 1.46}$

Table 4: Results on *CelebA* and *QM9* datasets with MR and Δm%. The results are averaged from 3 random seeds.

Method	CelebA		QM9	
	MR ↓	Δm% ↓	MR ↓	Δm% ↓
LS	9.98	4.15	13.36	177.6
SI	12.38	7.20	7.73	77.8
RLW	8.55	1.46	15.18	203.8
DWA	11.03	2.40	13.18	175.3
UW	9.20	3.23	9.45	108.0
MGDA	16.58	14.85	12.45	120.5
PCGrad	10.60	3.17	11.09	125.7
CAGrad	10.20	2.48	11.82	112.8
IMTL-G	7.85	0.84	10.36	77.2
Nash-MTL	8.05	2.84	6.27	62.0
FAMO	7.90	1.21	8.27	58.5
FairGrad	8.60	0.37	6.09	57.9
COST	9.55	0.93	7.09	58.3
Muon	9.20	$0.74_{\pm 0.44}$	11.00	$166.0_{\pm 15.43}$
APT-Muon	8.85	$0.03_{\pm 0.62}$	6.00	$72.9_{\pm 1.63}$
APT-CG	7.65	$1.12_{\pm 1.39}$	10.09	$108.9_{\pm 1.57}$
APT-FAMO	7.65	$1.04_{\pm 0.64}$	7.00	$55.5_{\pm 7.79}$
APT-Fair	7.20	$0.53_{\pm 1.55}$	4.55	$64.9_{\pm 3.55}$

[Yu et al., 2020], CAGrad from [Liu et al., 2021a], IMTL detailed in [Liu et al., 2021b], Nash-MTL from [Navon et al., 2022], FAMO described in [Liu et al., 2023], FairGrad from [Ban and Ji, 2024], and COST from [Zhou et al., 2025b].

Evaluation Metric: In addition to individual task performance, we incorporate a widely used aggregate metric, $\Delta m\%$ [Maninis et al., 2019]. This metric evaluates the overall performance degradation relative to independently trained models, which serve as single-task references. The

formal definition of $\boldsymbol{\Delta m}\%$ is:

$$\Delta m\% = \frac{1}{Q}\sum_{k=1}^{Q}(-1)^{\delta_k}\frac{(M_{m,k} - M_{b,k})}{M_{b,k}} \times 100 \tag{11}$$

where Q is the number of evaluation metrics, and $M_{m,k}$ and $M_{b,k}$ represent the value of the k-th metric for the compared multi-task method and the independent single-task baseline, respectively. The indicator δ_k is set to 1 if a higher value is preferred for metric M_k, and 0 otherwise. Furthermore, we report the Mean Rank (**MR**), which calculates the average ranking of each method across the reported metrics; a lower MR signifies more consistent performance across the benchmark.

Implementation Details: Adam optimizer settings. For the Cityscapes and NYUv2 datasets, we construct our model using the SegNet architecture [Badrinarayanan et al., 2017] and incorporate MTAN [Liu et al., 2019b] as the shared backbone. We train the models for 200 epochs using the Adam optimizer with an initial learning rate of 1.0×10^{-4}, which is halved after the 100th epoch. The batch size is set to 8 for Cityscapes and 2 for NYUv2. For CelebA, we utilize a 9-layer Convolutional Neural Network (CNN) as the backbone, followed by task-specific linear heads. The model is trained for 15 epochs with the Adam optimizer, using a learning rate of 3.0×10^{-4} and a batch size of 256. For QM9, our model is trained for 300 epochs with a batch size of 120. We initialize the learning rate at 1.0×10^{-3} and employ a "Reduce On Plateau" scheduler, which decreases the learning rate when validation performance stagnates, ensuring consistent convergence with prior work.

Muon Optimizer Settings. Regarding the Muon optimizer, we maintain consistency by sharing all architectural and training configurations with our Adam optimizer setup, with the exception of the learning rate. For CityScapes and NYUv2, the learning rate is set as 0.02; For CelebA, the learning rate is set as 5.e-3; For QM9, the learning rate is initialized as 5.e-3. We employ a "Reduce On Plateau" scheduler with a patience of 10 steps, decreasing the learning rate once the validation performance stagnates. Please refer to the Appendix Sec. A.2 for baselines' reproduction.

5.1 Main Results

We evaluate our framework on the four aforementioned MTL benchmarks. Specifically, we augment representative MTL approaches—including CAGrad (denoted as CG), FAMO, and FairGrad (denoted as Fair)—with the `APT` framework. We also include a Muon-based variant, `APT`-Muon, which combines adaptive momentum and LDP.

The results for the scene understanding benchmarks are presented in Table 2 and Table 3. As shown, `APT` generally augments the evaluated baselines, yielding considerable improvements in both MR and $\Delta m\%$. Notably, `APT`-Muon attains competitive performance on both benchmarks. This supports the practical compatibility of Muon with our framework; the separate contributions of adaptive momentum and LDP are examined below. Additionally, `APT`-Fair exhibits near-SOTA performance, further highlighting the critical importance of balancing instant and amortized de-conflicting. In many-task scenarios such as CelebA (40 tasks) and QM9 (11 tasks), `APT` continues to demonstrate strong augmentation effects across most baselines, as demonstrated in Table 4. For FairGrad, comparison with the reproduced baseline clarifies the mixed results: CelebA improves from 1.09 to 0.53, whereas QM9 degrades from 59.90 to 64.90. The latter coincides with noisier consecutive-gradient alignment, suggesting a less reliable temporal signal; it is not evidence that more tasks alone cause failure. `APT`-Muon remains competitive across these settings. Furthermore, `APT`-FAMO achieves SOTA results on the QM9 benchmark.

Against matched reimplementations, `APT` improves $\Delta m\%$ in 11 of the 12 CAGrad/FAMO/FairGrad–dataset pairs. The scene metrics yield 32 wins, 5 losses, and 2 ties across 39 displayed means (Appendix H.1); these counts are descriptive, not significance tests. These results collectively support the effectiveness of our proposed framework, demonstrating that `APT` is both a simple and highly effective solution for enhancing modern MTL optimization. We also provide comprehensive results on toy example in Appendix Sec. B.3.

5.2 Additional Analysis

Task-wise update compatibility. To complement the momentum traces, we evaluate Adam's final update direction. With FairGrad, the violated task–update fraction decreases from 0.0167 to 0.0000, while mean task–update cosine increases from 0.0731 to 0.1419. We further replay fixed $\beta = 0.9$

Table 5: Hidden task-wise violation rates on the same aggregate-gradient sequence. Reductions are percentage points with reported paired-bootstrap 95% intervals.

Dataset (tasks)	Fixed $\beta = 0.9$	APT	Reduction [95% CI]
Cityscapes (2)	1.88%	0.00%	1.88 [0.81, 3.23]
NYUv2 (3)	3.27%	0.25%	3.02 [1.51, 4.77]
CelebA (40)	72.92%	14.58%	58.33 [45.78, 72.92]

Table 6: Additional comparisons in $\Delta m\%$ (lower is better). Left: adaptive momentum with FairGrad. Right: Muon component ablation. Values are reported means.

Rule	Cityscapes	NYUv2
DEMON	11.04	2.99
AMM	4.40	-5.03
YellowFin	16.04	-2.71
MoCoGrad	6.25	-3.35
APT	**2.22**	**-6.04**

Adaptive	LDP	Cityscapes	NYUv2
		-2.56	-3.15
✓		-2.77	-3.74
	✓	-2.82	-3.53
✓	✓	**-3.09**	**-5.54**

on the same APT-generated aggregate sequence. A *hidden task-wise violation* occurs when the aggregate is non-conflicting and the update aligns with it, but the update conflicts with at least one task. Table 5 shows that APT reduces these events across 2, 3, and 40 tasks. On NYUv2/CelebA, the task-normalized violation fraction also decreases from $1.09\%/4.90\%$ to $0.08\%/0.42\%$. These diagnostics isolate update compatibility on common inputs, without establishing a causal link to final test performance. Tracking errors, conflict-adjusted associations, and protocol details are given in Appendix H.2.

Adaptive alternatives and component ablation. Table 6 compares adaptive rules with FairGrad and separates APT from LDP under Muon. APT has the lowest reported mean among the adaptive alternatives; AMM is a close comparator. Both Muon components improve the displayed means, with the combination performing best. Together with the fixed-momentum sweeps, these results support adaptation without establishing an equally tuned optimum or statistically significant interactions. APT's bounds remain fixed across datasets and methods.

6 Conclusion

In this work, we revisited the effectiveness of optimization-based MTL and identified a critical bottleneck: the disconnect between derived task gradients and the actual parameter updates. Our analysis revealed that this discrepancy prevents previous MTL frameworks from fully leveraging their intended learning dynamics. Furthermore, we highlighted a conditional task-rebalancing interpretation of the Muon optimizer and the importance of suitable gradient inputs for its orthogonalization process. To bridge these gaps, we introduced the `APT` framework, which integrates a simple adaptive momentum mechanism with a light direction preservation strategy. Extensive empirical evaluations across four mainstream datasets show that `APT` improves compatibility between advanced optimizers and MTL in most evaluated settings. Remaining limitations include noisy temporal signals, nonuniform task gains, and runtime overhead (Appendix D).

Acknowledgement

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). This research is also supported, in part, by the Centre of AI in Medicine (C-AIM), Jinan-NTU Green Technology Research Institute (GreenTRI), and Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore.

References

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

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *International Conference on Machine Learning*, pages 2715–2731. PMLR, 2024.

Linxiao Cao, Mianzimei Yang, Zhipeng Zhou, Hong Xie, Defu Lian, and Menglin Yang. Towards calibrated gradient-based multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5127–5136, 2026.

Heyan Chai, Zeyu Liu, Yongxin Tong, Ziyi Yao, Binxing Fang, and Qing Liao. Towards task-conflicts momentum-calibrated approach for multi-task learning. In *40th IEEE International Conference on Data Engineering*. IEEE, 2024.

John Chen, Cameron Wolfe, Zhao Li, and Anastasios Kyrillidis. Demon: Improved neural network training with momentum decay. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

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

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 27503–27516, 2021.

Falk Heuer, Sven Mantowsky, Saqib Bukhari, and Georg 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*, pages 997–1005, 2021.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021.

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

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

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

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019a.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1851–1860, 2019.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

Pierre Quinton and Valérian Rey. Jacobian descent for multi-objective optimization. *arXiv preprint arXiv:2406.16232*, 2024.

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

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

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

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

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

Xiaozhuang Song, Shun Zheng, Wei Cao, James Yu, and Jiang Bian. Efficient and effective multi-task grouping via meta learning on task combinations. *Advances in Neural Information Processing Systems*, 35:37647–37659, 2022.

Kristi Topollai and Anna Ewa Choromanska. Adaptive memory momentum via a model-based framework for deep learning optimization. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pages 4060–4068, 2026.

Peiyao Xiao, Chaosheng Dong, Shaofeng Zou, and Kaiyi Ji. Ldc-mtl: Balancing multi-task learning through scalable loss discrepancy control. *arXiv preprint arXiv:2502.08585*, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Jian Zhang and Ioannis Mitliagkas. YellowFin and the art of momentum tuning. In *Proceedings of Machine Learning and Systems*, 2019.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. *arXiv preprint arXiv:2410.20008*, 2024.

Zhipeng Zhou, Liu Liu, Peilin Zhao, and Wei Gong. Injecting imbalance sensitivity for multi-task learning. *arXiv preprint arXiv:2503.08006*, 2025a.

Zhipeng Zhou, Ziqiao Meng, Pengcheng Wu, Peilin Zhao, and Chunyan Miao. Continual optimization with symmetry teleportation for multi-task learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Zhipeng Zhou, Ziqiao Meng, Pengcheng Wu, Peilin Zhao, and Chunyan Miao. Exploring tradeoffs through mode connectivity for multi-task learning. *Advances in Neural Information Processing Systems*, 38:81268–81285, 2025c.

The appendix provides implementation details (Appendix A), additional results (B), optimizer background and limitations (C–D), alignment and auxiliary convergence proofs (E–F), the conditional Muon proof and spectral diagnostics (G), and supplementary matched comparisons, update diagnostics, and overhead measurements (H).

A Implementation

A.1 Toy Example

We adopt the similar toy optimization example introduced in [Liu et al., 2023], which involves two objectives that consider the variables $\boldsymbol{\vartheta} = (\vartheta_1, \vartheta_2)$:

$$\begin{aligned}
\mathcal{L}_1(\boldsymbol{\vartheta}) &= 0.5 \cdot (c_1(\boldsymbol{\vartheta}) f_1(\boldsymbol{\vartheta}) + c_2(\boldsymbol{\vartheta}) g_1(\boldsymbol{\vartheta})) \\
\mathcal{L}_2(\boldsymbol{\vartheta}) &= c_1(\boldsymbol{\vartheta}) f_2(\boldsymbol{\vartheta}) + c_2(\boldsymbol{\vartheta}) g_2(\boldsymbol{\vartheta}) \qquad \text{where} \\
f_1(\boldsymbol{\vartheta}) &= \log(\max(|0.5(-\vartheta_1 - 7) - \tanh(\vartheta_2)|, 5.e-6)) + 6, \\
f_2(\boldsymbol{\vartheta}) &= \log(\max(|0.5(-\vartheta_1 + 3) - \tanh(\vartheta_2) + 2|, 5.e-6)) + 6, \\
g_1(\boldsymbol{\vartheta}) &= ((-\vartheta_1 + 7)^2 + 0.1 * (-\vartheta_2 - 8)^2)/10 - 20, \\
g_2(\boldsymbol{\vartheta}) &= ((-\vartheta_1 - 7)^2 + 0.1 * (-\vartheta_2 - 8)^2)/10 - 20, \\
c_1(\boldsymbol{\vartheta}) &= \max(\tanh(0.5 * \vartheta_2), 0), \\
c_2(\boldsymbol{\vartheta}) &= \max(\tanh(-0.5 * \vartheta_2), 0).
\end{aligned} \tag{12}$$

The initial points in Figure 2 of the main text are $\{(-8.5, 7.5), (-8.5, 5), (0, 0), (9, 9), (10, -8)\}$, while those in Table 10 are $\{(-8.5, 7.5), (0, 8), (5, 9)\}$.

A.2 Baselines

All baselines that `APT` chooses to augment with are re-implemented according to their official implementations. Here we would like to provide our reproduced results to further demonstrate the improvements brought by `APT`. The results are presented in Table 7,and 8. Published and reproduced baseline scores can differ, so augmentation gains should be assessed against the matched reimplementations. The complete aggregate comparison is in Appendix H.1. Besides, all experiments are carried out on NVIDIA 4090/A100 GPUs.

B More Results

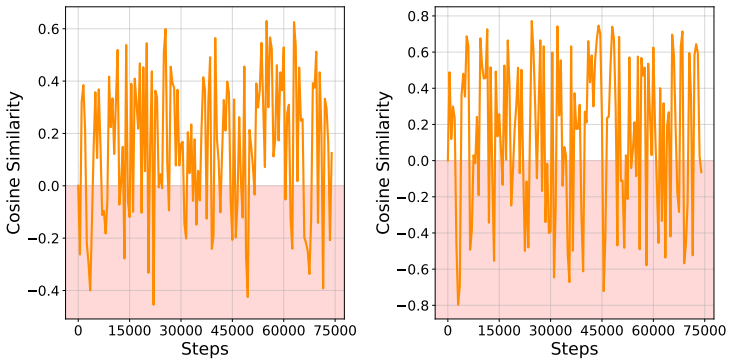
B.1 Consecutive-Gradient Directional Variation

Here we take two representative MTL approaches, CAGrad and FairGrad, to examine directional changes during training. Specifically, we track their cosine similarity between the gradients from adjacent steps and present the results in Figure 4. As shown, many training steps exhibit low alignment. This can reflect minibatch noise, task-weight changes, or parameter movement, and does not by itself identify curvature or task conflict.

B.2 Preliminary Discussion on the Muon's Input Gradient

In this subsection, we want to answer a key question: What Kind of MTL is Suitable for Muon? Mathematically, Muon acts as a basis enhancer by orthogonalizing the update matrix, and its effectiveness therefore depends on the directions of the input gradients provided by the MTL algorithm. However, a higher rank is not unconditionally beneficial. Since Muon forces all non-zero singular values to unity ($\sigma_i \to 1$), it may amplify noise by promoting weak or stochastic directions to the same scale as dominant task signals. Consequently, an ideal MTL partner for Muon should preserve a sufficiently high effective rank [Roy and Vetterli, 2007] while allowing Muon to handle the primary balancing through orthogonalization.

As shown in Table 1, Muon performs well with CAGrad and Nash-MTL, but less favorably with MGDA and FairGrad. One possible explanation involves Muon's spectral orthogonalization, which preserves dominant directions while flattening the singular value spectrum, thereby emphasizing directional geometry over magnitude-based weighting. CAGrad and Nash-MTL may align with this



(a) CAGrad. (b) FairGrad.

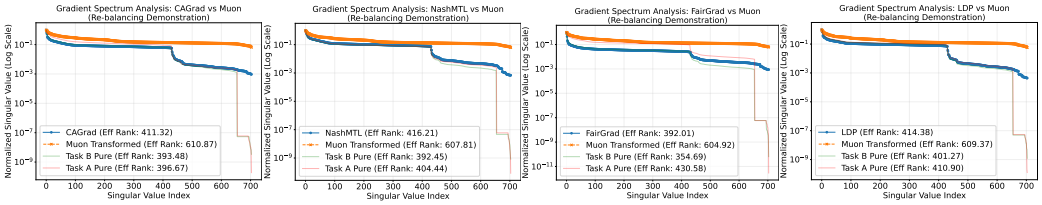
Figure 4: Similarity between gradients from adjacent steps.

Table 7: **Scene understanding** (*CityScapes*, 2 tasks). All results are reproduced averaged over 3 random seeds.

Method	Segmentation ↑		Depth ↓		$\Delta\mathbf{m}\%$ ↓
	mIoU	Pix.	Abs.	Rel.	
MGDA	71.78	92.39	0.0134	32.72	7.29
APT-MGDA	68.99	91.82	0.0134	29.31	5.16
CAGrad	75.38	93.57	0.0142	38.38	12.30
APT-CAGrad	75.79	93.65	0.0129	38.13	9.31
FAMO	73.50	93.06	0.0152	33.03	10.39
APT-FAMO	75.11	93.48	0.0138	35.39	9.03
FairGrad	74.61	93.36	0.0131	31.47	4.23
APT-FairGrad	74.29	93.25	0.0125	30.36	2.22

mechanism, as they seek central compromise directions and encode task trade-offs mainly through directional structure, resulting in smoother and less degenerate spectra. In contrast, MGDA and FairGrad may be more sensitive to loss of magnitude information under spectral flattening; these observations do not isolate the cause of their performance differences.

Further evidence is provided in Figure 5. FairGrad exhibits the lowest effective rank, while CAGrad and Nash-MTL maintain higher-rank structures. We also provide the effective rank observation for our approach, LDP. As shown in Figure 5, LDP has a higher input effective rank than FairGrad but a lower one than Nash-MTL in this example. These diagnostics motivate soft aggregation without proving rank preservation or universal task balancing.



(a) CAGrad. (b) Nash-MTL. (c) FairGrad. (d) LDP.

Figure 5: Effective rank comparison.

Table 8: **Scene understanding** (*NYUv2*, 3 tasks). All results are reproduced averaged over 3 random seeds. The baselines' results are reported from FairGrad's implementation.

Method	Segmentation ↑		Depth ↓		Surface Normal					$\Delta\mathbf{m}\%$ ↓
	mIoU	Pix. Acc.	Abs. Err.	Rel. Err.	Angle Distance ↓		Within t° ↑			
					Mean	Median	11.25	22.5	30	
MGDA	31.87	60.76	0.60	0.23	24.74	19.17	30.18	57.44	69.65	-0.81
APT-MGDA	31.75	60.78	0.60	0.24	24.72	19.06	30.40	57.61	69.69	-0.98
CAGrad	40.01	66.38	0.55	0.22	26.66	22.38	24.51	50.75	64.47	1.30
APT-CAGrad	40.51	66.32	0.53	0.22	25.77	20.65	27.24	54.25	67.15	-2.67
FAMO	36.43	63.37	0.56	0.23	24.97	19.20	30.06	57.34	69.37	-3.49
APT-FAMO	37.66	64.57	0.54	0.23	24.87	19.11	30.12	57.50	69.58	-4.35
FairGrad	38.54	64.85	0.56	0.23	24.46	18.90	30.64	58.09	70.25	-4.91
APT-FairGrad	38.97	65.07	0.54	0.22	24.44	18.85	30.84	58.12	70.22	-6.04

Table 9: Analysis on β across various MTL approaches and advanced optimizers on *NYUv2*. AdamW is set with weight decay as 1.e-4 to distinguish with Adam. All results are average on 3 random seeds. The best scores are provided in gray. The baselines' results are reported from our implementation.

β	MGDA			CAGrad		
	Adam	AdamW	Muon	Adam	AdamW	Muon
default	1.29 ± 0.27	-1.04 ± 0.34	-4.57 ± 0.48	-1.81 ± 1.76	-1.27 ± 0.34	-9.78 ± 0.68
0.6	-1.22 ± 0.33	-0.52 ± 0.49	-0.94 ± 0.51	-1.86 ± 1.37	-3.38 ± 0.54	-1.78 ± 1.66
0.2	-0.35 ± 1.22	-0.88 ± 0.45	5.19 ± 0.81	-3.17 ± 0.87	-2.67 ± 1.71	4.91 ± 3.27
0.0	-0.44 ± 0.16	-0.38 ± 0.23	7.40 ± 0.95	-3.37 ± 0.75	-1.68 ± 0.80	6.92 ± 2.04
β	Nash-MTL			FairGrad		
	Adam	AdamW	Muon	Adam	AdamW	Muon
default	4.32 ± 1.96	3.95 ± 0.28	-12.43 ± 0.43	-4.74 ± 0.47	-4.16 ± 0.48	-13.23 ± 0.72
0.6	3.78 ± 3.30	3.47 ± 0.86	3.04 ± 0.64	-4.69 ± 0.64	-2.90 ± 1.50	-5.73 ± 1.03
0.2	4.32 ± 2.20	4.07 ± 2.63	11.94 ± 0.20	-5.39 ± 0.34	-4.36 ± 1.20	1.98 ± 1.21
0.0	4.99 ± 1.46	5.30 ± 3.24	13.23 ± 2.00	-4.55 ± 0.58	-3.55 ± 0.79	3.00 ± 0.76

B.3 Toy Example

To further illustrate the enhanced convergence properties of APT, we present additional comparative synthetic examples under various imbalance scenarios in Table 10. The results reveal that APT successfully augments nearly all baselines, enabling them to reach the Pareto front, although convergence may still fail from specific starts. These findings underscore the effectiveness of APT's balancing mechanism in navigating complex multi-task landscapes.

B.4 Augment on the Most Related MTL Baseline

As discussed in the related work section, UPGrad is the most closely related method to our approach. However, our method is orthogonal to it. To demonstrate compatibility, we augment UPGrad with APT and conduct experiments on Cityscapes, with results reported in Table 11. As shown, APT improves the aggregate metric for UPGrad, although the absolute depth error changes from 0.0146 to 0.0147.

C Introduction of Optimizers

In this section, we briefly review the optimization algorithms discussed in this work: Adam, AdamW, and the recently proposed Muon optimizer.

Table 10: Comparison of MTL optimization methods with and w/o APT on synthetic two-task benchmark. (a_1, a_2) denotes $a_1 * \mathcal{L}_1 + a_2 * \mathcal{L}_2$.

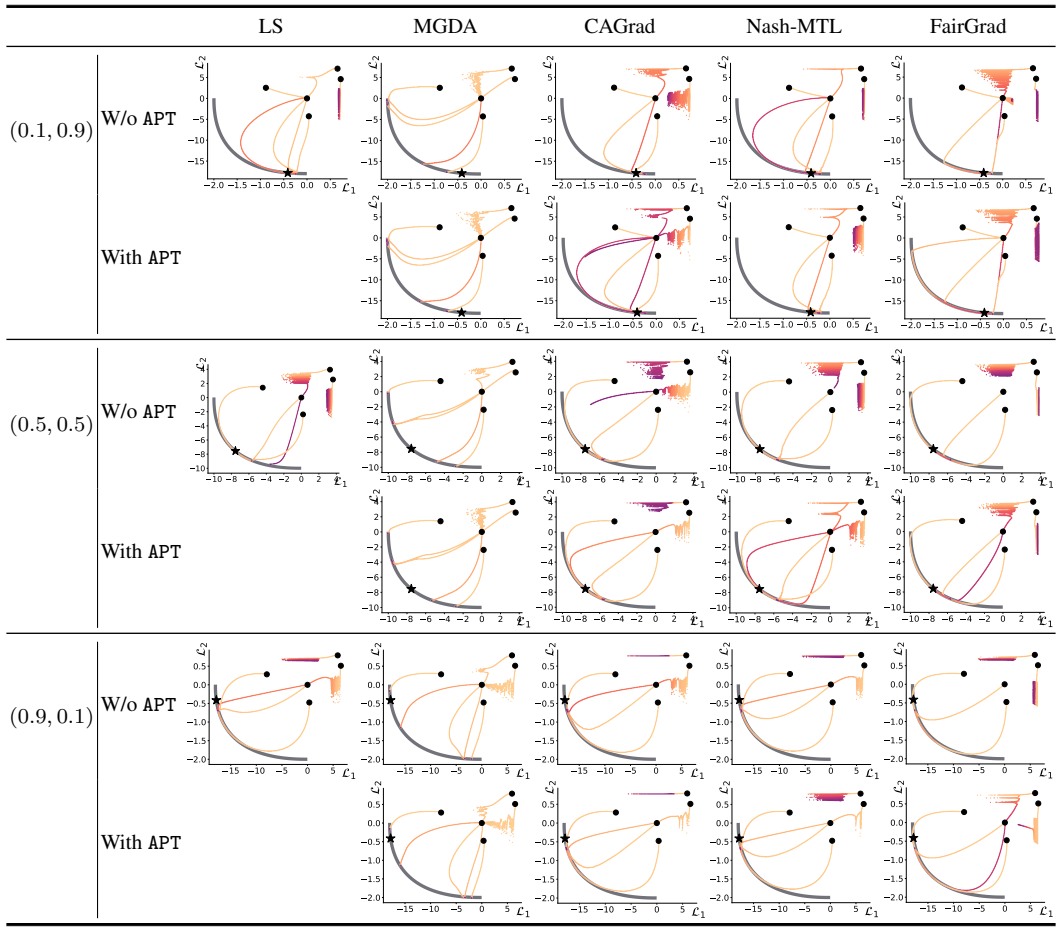


Table 11: **Scene understanding** (*CityScapes*, 2 tasks). All results are reproduced averaged over 3 random seeds.

Method	Segmentation ↑		Depth ↓		$\Delta\mathbf{m}\%$ ↓
	mIoU	Pix.	Abs.	Rel.	
UPGrad	75.88	93.63	0.0146	44.22	18.29
APT-UPGrad	75.98	93.63	0.0147	43.31	17.69

Adam. Adam is a first-order gradient-based optimizer that adapts learning rates for individual parameters. It maintains exponential moving averages of the gradients (m_t) and the squared gradients (v_t):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

The parameters θ are updated using bias-corrected estimates ($\hat{m}_t, \hat{v}_t$):

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

While effective, Adam's coupling of weight decay with the adaptive learning rate can lead to suboptimal regularization.

AdamW. AdamW addresses the regularization issue in Adam by decoupling the weight decay from the gradient update. Instead of adding the L_2 penalty to the gradient g_t, AdamW applies it directly to

the parameters during the update step:

$$\theta_t = \theta_{t-1} - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_{t-1} \right)$$

This modification ensures that weight decay remains effective even when the adaptive learning rate is small, significantly improving generalization performance.

Muon. Unlike Adam and AdamW, which operate coordinate-wise, Muon is a matrix-specific optimizer designed to balance updates across the spectral directions of neural network layers. It treats the internal layers as 2D matrices and replaces the coordinate-wise scaling with an orthogonalization step.Muon first computes a standard momentum buffer M_t. Instead of dividing by variance, it projects M_t onto the manifold of orthogonal matrices using Newton-Schulz iterations, effectively normalizing the singular values of the update matrix. The ideal polar factor flattens nonzero singular values; finite Newton–Schulz iterations and implementation conventions approximate this transformation. Singular directions are not task identities, and the task-balancing interpretation requires the conditions in Theorem 3.

D Limitation

While this paper introduces a simple and effective framework for augmenting current MTL baselines, it is important to acknowledge certain limitations. First, our approach serves as an initial exploration into the fundamental challenge of balancing instant and amortized de-conflicting in MTL. While our adaptive momentum strategy provides a robust baseline, more sophisticated architectural or algorithmic designs specifically tailored to high-dimensional task landscapes may further enhance stability. The regression of `APT`-Fair on QM9 highlights the trade-off between gradient denoising and temporal responsiveness. A low consecutive-gradient cosine can reflect noise, and neither the auxiliary tracking bounds nor reduced instantaneous violations guarantee improved final task metrics. Second, while we have focused primarily on the synergy between MTL and the Muon and Adam optimizers, extending this discussion to other advanced optimization frameworks—such as Lion, Shampoo, or Federated optimization methods—could yield broader insights into the generalizability of our claims.

E Proofs of the Alignment and Transition Identities

E.1 Amortized alignment identity

For fixed β, $m_0 = 0$, and a locally constant reference gradient $\bar{g}_i$, since $m_t = (1-\beta)\sum_{k=1}^{t} \beta^{t-k} g_{mtl,k}$, consider the inner product between m_t and $\bar{g}_i$:

$$\begin{aligned} \langle \bar{g}_i, m_t \rangle &= \langle \bar{g}_i, (1-\beta) \sum_{k=1}^{t} \beta^{t-k} g_{mtl,k} \rangle \\ &= (1-\beta) \sum_{k=1}^{t} \beta^{t-k} \langle \bar{g}_i, g_{mtl,k} \rangle \end{aligned} \tag{13}$$

On the other hand, we also have $\langle \bar{g}_i, g_{mtl,k} \rangle = \langle \hat{g}_{i,k}, g_{mtl,k} \rangle - \langle \epsilon_{i,k}, g_{mtl,k} \rangle$. By substituting this equation into Eqn.(13), we finally obtain the Eqn.(2).

Zero-mean noise alone does not imply $\mathbb{E}\langle \epsilon_{i,k}, g_{mtl,k} \rangle = 0$: taking $g_{mtl,k} = \epsilon_{i,k}$ gives $\mathbb{E}\|\epsilon_{i,k}\|^2 > 0$ for nondegenerate noise. Conditional orthogonality, $\mathbb{E}[\langle \epsilon_{i,k}, g_{mtl,k} \rangle \mid \mathcal{F}_{k-1}] = 0$, suffices for cancellation. Otherwise positive expected alignment requires the expected step-wise term to dominate the expected noise correction. An illustrative mean leading-to-correction ratio of 9.38 supports dominance in the measured setting, but is not a universal bound. Nonzero initial momentum or a drifting reference gradient introduces additional terms.

E.2 Transition identity

Under the constant expected post-transition projection assumed in the main text, at time $t = T + \Delta t$, the momentum can be decomposed as:

$$m_t = \underbrace{\beta^{\Delta t+1} m_{T-1}}_{\text{Stale Momentum}} + \underbrace{(1-\beta)\sum_{k=T}^{t} \beta^{t-k} g_{mtl,k}}_{\text{Fresh Updates}} \tag{14}$$

Taking expectation and inner product with the new gradient:

$$\langle \bar{g}_i^{(new)}, \mathbb{E}[m_t] \rangle = \beta^{\Delta t+1} \langle \bar{g}_i^{(new)}, \mathbb{E}[m_{T-1}] \rangle + (1-\beta^{\Delta t+1}) \langle \bar{g}_i^{(new)}, \mathbb{E}[\bar{g}_{mtl}^{(new)}] \rangle \tag{15}$$

Substituting:

$$\langle \bar{g}_i^{(new)}, \mathbb{E}[m_t] \rangle = \beta^{\Delta t+1}(-C) + (1-\beta^{\Delta t+1}) S \tag{16}$$

To avoid Pareto violation $\langle \bar{g}_i^{(new)}, \mathbb{E}[m_t] \rangle \geq 0$, which yields $(1-\beta^{\Delta t+1})S \geq \beta^{\Delta t+1} C$. By rearranging, we finally obtain the Eqn. (4).

F Auxiliary Tracking and Surrogate-Stationarity Analysis

The results in this appendix concern generic momentum tracking and conditional inexact descent. They can also be instantiated in single-task optimization. They are not presented as interaction-specific MTL convergence theorems, and do not by themselves prove convergence of practical stochastic APT to the Pareto front.

F.1 Tracking error

Theorem 1 (Finite-horizon tracking bound)**.** *Let $m_t = \beta_t m_{t-1} + (1-\beta_t) G_t$, $\delta_t = m_t - G_t$, and $0 \leq \beta_t \leq \beta_{\max} < 1$. Choose $\gamma > 0$ such that $c = (1+\gamma)\beta_{\max}^2 < 1$. For any $T \geq 1$,*

$$\frac{1}{T}\sum_{t=1}^{T} \|\delta_t\|^2 \leq \frac{1+1/\gamma}{1-c} \frac{1}{T} \sum_{t=1}^{T} \beta_t^2 \|G_t - G_{t-1}\|^2 + \frac{c\|\delta_0\|^2}{T(1-c)}. \tag{17}$$

Proof. Equation (7) and Young's inequality yield

$$\|\delta_t\|^2 \leq (1+\gamma)\beta_t^2 \|\delta_{t-1}\|^2 + (1+1/\gamma)\beta_t^2 \|G_t - G_{t-1}\|^2 \tag{18}$$

$$\leq c\|\delta_{t-1}\|^2 + (1+1/\gamma)\beta_t^2 \|G_t - G_{t-1}\|^2. \tag{19}$$

Summing over t gives $(1-c)\sum_{t=1}^{T} \|\delta_t\|^2 \leq c\|\delta_0\|^2 - c\|\delta_T\|^2 + (1+1/\gamma)\sum_{t=1}^{T} \beta_t^2 \|G_t - G_{t-1}\|^2$. Dropping the nonpositive terminal term and dividing by $T(1-c)$ proves the claim. The long-run average bound follows by taking a limsup. □

The bound is pathwise and does not require independence of β_t and G_t. It explains the role of momentum in the injection term; it does not establish that the cosine rule minimizes that term, nor does it compare two independently evolving training trajectories.

F.2 A task-wise consequence, without an interaction model

If $\langle g_{i,t}, G_t \rangle \geq 0$ and $\|g_{i,t}\| \leq B$ for every task, then

$$[-\langle g_{i,t}, m_t \rangle]_+ \leq |\langle g_{i,t}, \delta_t \rangle| \leq B\|\delta_t\|. \tag{20}$$

Define $V_t = K^{-1}\sum_{i=1}^{K} [-\langle g_{i,t}, m_t \rangle]_+$. Averaging over tasks gives $V_t \leq B\|\delta_t\|$. Averaging over steps and applying Cauchy–Schwarz gives

$$\frac{1}{T}\sum_{t=1}^{T} V_t \leq B\sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\delta_t\|^2}. \tag{21}$$

Combining with Theorem 1 bounds the average *magnitude* of negative first-moment projections. It does not bound their event count without additional margin assumptions, and it does not model pairwise task interactions. For a preconditioned update, one would also need a non-conflicting preconditioned aggregate and bounds on the transformed task gradients; ordinary common descent alone is insufficient.

F.3 Assumptions for a surrogate-descent bound

For this subsection, consider $\theta_{t+1} = \theta_t - \tilde{\eta}_t \hat{H}_t^{-1} m_t$, where $\tilde{\eta}_t = \eta/q_t$, $q_t = 1 - \prod_{j=1}^{t} \beta_j$, and $\eta > 0$ is constant. Assume:

1. F is L-smooth and bounded below by F_*.
2. For constants $\mu, M > 0$ and residual $\epsilon \geq 0$,
$$\langle \nabla F(\theta_t), \hat{H}_t^{-1} G_t \rangle \geq \mu \|\nabla F(\theta_t)\|^2 - \epsilon, \qquad \|\hat{H}_t^{-1} G_t\| \leq M \|\nabla F(\theta_t)\|. \tag{22}$$
3. $\hat{H}_t$ is symmetric positive definite with $c_l I \preceq \hat{H}_t \preceq c_u I$, for $0 < c_l \leq c_u < \infty$.

These are explicit additional assumptions, not automatic consequences of applying an arbitrary MTL aggregator to Adam. In particular, the existence of a single suitable smooth surrogate is restrictive. The residual ϵ measures failure of the stated surrogate-descent inequality; it should not be identified with every empirical task-conflict statistic. For Adam, bounded input gradients and a positive denominator offset can ensure bounded diagonal preconditioners, but do not establish the surrogate inequalities.

F.4 Conditional ergodic/liminf stationarity

Let $\eta_{\max} = \eta/(1-\beta_{\max})$, so $\eta \leq \tilde{\eta}_t \leq \eta_{\max}$. Choose η small enough that $A = \mu/2 - L\eta_{\max} M^2 > 0$ and define $B_0 = 1/(2\mu c_l^2) + L\eta_{\max}/c_l^2$.

Theorem 2 (Surrogate-stationarity bound)**.** *Under the preceding assumptions,*

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla F(\theta_t)\|^2 \leq \frac{F(\theta_1) - F_*}{\eta A T} + \frac{\eta_{\max} B_0}{\eta A} \frac{1}{T}\sum_{t=1}^{T} \|\delta_t\|^2 + \frac{\eta_{\max}}{\eta A}\epsilon. \tag{23}$$

Consequently the liminf of $\|\nabla F(\theta_t)\|^2$ is bounded by the limsup average tracking term and the residual term on the right.

Proof. Smoothness and $m_t = G_t + \delta_t$ give

$$\begin{aligned} F(\theta_{t+1}) - F(\theta_t) \leq & -\tilde{\eta}_t \langle \nabla F, \hat{H}_t^{-1} G_t \rangle - \tilde{\eta}_t \langle \nabla F, \hat{H}_t^{-1} \delta_t \rangle \\ & + \frac{L\tilde{\eta}_t^2}{2} \|\hat{H}_t^{-1}(G_t + \delta_t)\|^2. \end{aligned} \tag{24}$$

The cross term is bounded by $\tilde{\eta}_t \mu \|\nabla F\|^2/2 + \tilde{\eta}_t \|\delta_t\|^2/(2\mu c_l^2)$. Also, $\|\hat{H}_t^{-1}(G_t + \delta_t)\|^2 \leq 2M^2\|\nabla F\|^2 + 2\|\delta_t\|^2/c_l^2$. Combining these bounds with Eq. (22) yields

$$F(\theta_{t+1}) - F(\theta_t) \leq -\eta A \|\nabla F(\theta_t)\|^2 + \eta_{\max} B_0 \|\delta_t\|^2 + \eta_{\max}\epsilon. \tag{25}$$

Telescoping, lower-bounding $F(\theta_{T+1})$ by F_*, and dividing by $\eta A T$ proves Eq. (23). The liminf conclusion follows from the inequality between a sequence's liminf and the limsup of its Cesaro averages. □

This establishes an average-gradient bound, not convergence of all iterates to a stationary neighborhood. Nor does stationarity of an unspecified surrogate automatically imply Pareto stationarity of the task losses. Such an interpretation requires an explicit relation between ∇F and a convex combination of task gradients (for example, a fixed positive weighted sum of the task losses). That relation is not supplied by the tracking theorem alone.

F.5 Diminishing-step variant: weighted stationarity, not full-sequence convergence

Assume additionally that $G_t = G(\theta_t)$ is a deterministic L_G-Lipschitz map, $\|G_t\| \le G_{\max}$, $m_0 = 0$, and the surrogate residual is zero. Take $\eta_t = \eta_0/\sqrt{t}$ with sufficiently small $\eta_0 > 0$. These assumptions exclude persistent independent minibatch fluctuations; a diminishing learning rate alone does not remove such gradient noise.

The bounded moment and preconditioner imply $\|\theta_{t+1} - \theta_t\| \le \eta_t G_{\max}/[(1-\beta_{\max})c_l] = O(t^{-1/2})$. Lipschitz continuity therefore gives $\|G_t - G_{t-1}\|^2 = O(t^{-1})$. The tracking recurrence implies

$$\|\delta_t\|^2 \le c^t \|\delta_0\|^2 + C_1 \sum_{j=1}^{t} \frac{c^{t-j}}{j} = O(t^{-1}). \tag{26}$$

To see the last step, split the sum at $t/2$: the early terms decay geometrically, while the late terms are at most $2/[t(1-c)]$. The descent proof above, with constants uniform over time, then yields

$$\sum_{t=1}^{T} \eta_t A \|\nabla F(\theta_t)\|^2 \le F(\theta_1) - F_* + B_1 \sum_{t=1}^{T} \eta_t \|\delta_t\|^2. \tag{27}$$

The last series is bounded because its terms are $O(t^{-3/2})$, whereas $\sum_{t=1}^{T} \eta_t = \Theta(\sqrt{T})$. Hence

$$\min_{1\le t\le T} \|\nabla F(\theta_t)\|^2 \le \frac{\sum_{t=1}^{T} \eta_t \|\nabla F(\theta_t)\|^2}{\sum_{t=1}^{T} \eta_t} = O(T^{-1/2}). \tag{28}$$

The justified conclusions are weighted ergodic stationarity and a zero liminf. This argument does not by itself establish full-sequence gradient convergence or exact Pareto convergence for stochastic MTL.

G Proof and Diagnostics for Theorem 3

Here G_t denotes the matrix input to the ideal polar transformation; in practical Muon this input is momentum-based. Let the SVD of G_t be $G_t = U\Sigma V^\top = \sum_{k=1}^{r} \sigma_k u_k v_k^\top$, and define orthogonal bases $E_k = u_k v_k^\top$. Then the ideal polar update can be written as $O_t = \sum_{k=1}^{r} E_k$. Each task gradient admits the decomposition $g_i = \sum_{k=1}^{r} \alpha_{i,k} E_k + g_{i,\perp}, \quad \alpha_{i,k} = \langle g_i, E_k \rangle_F$.

Theorem 3 (Conditional Task Projection Rebalancing of Muon)**.** *Let $P_i(\cdot) = \langle \cdot, g_i \rangle_F$ denote the projection onto task i. Assume that singular values are ordered as $\sigma_1 \ge \cdots \ge \sigma_r > 0$. Then, for any pair of tasks (i, j) satisfying: (1) task i, $\alpha_{i,1} \ge \cdots \ge \alpha_{i,r} \ge 0$, (2) task j, $0 \le \alpha_{j,1} \le \cdots \le \alpha_{j,r}$, with $\sum_k \alpha_{j,k} > 0$, the following inequality holds: $\frac{P_i(G_t)}{P_j(G_t)} \ge \frac{P_i(O_t)}{P_j(O_t)}$.*

We analyze the projection of both G_t and O_t onto task gradient g_i:

$$P_i(G_t) = \langle G_t, g_i \rangle_F = \sum_{k=1}^{r} \sigma_k \alpha_{i,k},$$

$$P_i(O_t) = \langle O_t, g_i \rangle_F = \sum_{k=1}^{r} \alpha_{i,k}.$$

Let $\bar{\sigma} = \frac{1}{r} \sum_{k=1}^{r} \sigma_k$ denote the mean singular value.

Task i. For task i with $\alpha_{i,1} \ge \cdots \ge \alpha_{i,r} \ge 0$, applying Chebyshev's sum inequality yields:

$$\sum_{k=1}^{r} \sigma_k \alpha_{i,k} \ge \frac{1}{r} \left(\sum_{k=1}^{r} \sigma_k \right) \left(\sum_{k=1}^{r} \alpha_{i,k} \right) = \bar{\sigma} P_i(O_t).$$

Task j. For task j with $0 \le \alpha_{j,1} \le \cdots \le \alpha_{j,r}$, Chebyshev's inequality gives:

$$\sum_{k=1}^{r} \sigma_k \alpha_{j,k} \le \bar{\sigma} P_j(O_t).$$

Since $\sum_k \alpha_{j,k} > 0$ and every $\sigma_k > 0$, both denominators are positive. Combining these inequalities, we find:

$$\frac{P_i(G_t)}{P_j(G_t)} \ge \frac{\bar{\sigma} P_i(O_t)}{\bar{\sigma} P_j(O_t)} = \frac{P_i(O_t)}{P_j(O_t)}.$$

Theorem 3 shows that the orthogonalized update reduces this ordered projection ratio under the stated loading assumptions. Contraction toward equality additionally requires $P_i(O_t) \ge P_j(O_t) > 0$. Singular directions are not task identities, and the sign and ordering conditions need not hold universally. An illustrative task–spectrum diagnostic follows.

G.1 Illustrative task–spectrum diagnostics

On Cityscapes, the reported analysis uses 149 checkpoints and 596 layer-level SVD samples. An effective spectral scale for task i is

$$\lambda_i^{\text{eff}} = \frac{\sum_k \alpha_{i,k}^2 \sigma_k}{\sum_k \alpha_{i,k}^2}. \tag{29}$$

Table 12: Illustrative Cityscapes task–spectrum summary.

Diagnostic	Semantic segmentation	Depth
Effective spectral scale	0.013924	0.013640
Bottom-band loading	0.74%	1.06%

Depth has a lower effective spectral scale and greater bottom-band exposure; a within-layer mode-permutation test reports $p = 0.000999$. These summaries suggest nonrandom task loadings in this example. Squared-loading averages, however, do not verify the theorem's full pointwise monotonicity and sign assumptions. The association neither demonstrates causal improvements in task trade-offs nor establishes a universal Muon property.

Table 13: Matched-implementation comparison: $\Delta m\%$ (lower is better). Bold values improve on the corresponding reproduced baseline. The win/loss summary counts method–dataset aggregate comparisons, not individual task metrics or significance tests.

Method	Cityscapes	NYUv2	CelebA	QM9
CAGrad	12.30	1.30	2.16	112.30
APT-CAGrad	**9.31**	**-2.67**	**1.12**	**108.90**
FAMO	10.39	-3.49	1.15	68.60
APT-FAMO	**9.03**	**-4.35**	**1.04**	**55.50**
FairGrad	4.23	-4.91	1.09	59.90
APT-FairGrad	**2.22**	**-6.04**	**0.53**	64.90
Wins / losses	3 / 0	3 / 0	3 / 0	2 / 1

H Supplementary Comparisons and Diagnostics

H.1 Matched comparisons and metric-level trade-offs

Table 13 compares official-code reimplementations and their APT counterparts under the same experimental settings. The broader non-APT MTL rows in the main benchmark tables are published results; they should not replace these matched baselines when measuring augmentation gains. The static momentum sweeps are a separate experiment set, so their default values can differ slightly from this matched comparison. No equally budgeted, validation-selected search is claimed. The matched scene-understanding tables yield 32 metric wins, 5 losses, and 2 ties for the three primary methods: CAGrad has 4/0/0 on Cityscapes and 7/1/1 on NYUv2; FAMO has 3/1/0 and 8/0/1; FairGrad has 2/2/0 and 8/1/0. These counts use displayed rounded means, not statistical significance. Complete matched per-attribute/target values for CelebA and QM9 are not reported here. For example, APT-FairGrad improves both Cityscapes depth errors but reduces both segmentation scores.

H.2 Update diagnostics

Let $\mathcal{C}_t = \{d : \langle g_{i,t}, d\rangle \geq 0 \ \forall i\}$ denote the common-descent cone. The hidden task-wise violation used in the replay comparison is

$$G_t \in \mathcal{C}_t, \qquad \langle G_t, d_t\rangle > 0, \qquad d_t \notin \mathcal{C}_t. \tag{30}$$

Thus agreement with the aggregate can mask disagreement with an individual task. The stale-momentum phenomenon itself is not exclusive to MTL. The pre-aggregation task-conflict statistic used in the regression is

$$C_t = \frac{2}{K(K-1)} \sum_{i<j} \max\{0, -\cos(g_{i,t}, g_{j,t})\}. \tag{31}$$

The reported CelebA reduction of 58.33 percentage points is computed before rounding; subtraction of the displayed rounded rates yields 58.34. The intervals in the main text are paired replay intervals, not across-seed uncertainty on final task performance. For a positive descent direction d_t, define the task-violation fraction $U_t = K^{-1} \sum_i \mathbf{1}\{\langle g_{i,t}, d_t\rangle < 0\}$ and the worst-task cosine $W_t = \min_i \cos(g_{i,t}, d_t)$. Hidden violation is the event in Eq. (30); it additionally requires a non-conflicting aggregate and positive aggregate–update alignment. These event rates and task fractions are different quantities and should not be compared as if they had the same denominator.

The reported task-projection CV uses normalized task projections $p_{i,t} = \langle g_{i,t}, d_t\rangle / \|g_{i,t}\|$:

$$\mathrm{CV}_t = \frac{\mathrm{Std}_i(p_{i,t})}{|\,\mathrm{Mean}_i(p_{i,t})| + \varepsilon}. \tag{32}$$

It measures relative dispersion, and can be sensitive when the mean projection is close to zero. The tracking analysis concerns $\delta_t = m_t - G_t$, while task-alignment diagnostics concern the update direction. This distinction is essential when interpreting the measurements alongside the auxiliary bounds.

Table 14: Controlled replay on the same aggregate sequence. H: hidden task-wise violation rate; U: task-violation fraction; W: worst-task update cosine; CV: task-projection coefficient of variation. A dash means not reported for that replay diagnostic. Rates are percentages.

Dataset (tasks)	Update	$H\downarrow$	$U\downarrow$	$W\uparrow$	CV$\downarrow$
Cityscapes (2)	Fixed-β replay	1.88	–	–	–
	APT	**0.00**	–	–	–
NYUv2 (3)	Fixed-β replay	3.27	1.09	0.03594	0.54531
	APT	**0.25**	**0.08**	**0.07706**	**0.50566**
CelebA (40)	Fixed-β replay	72.92	4.90	0.00072	0.93783
	APT	**14.58**	**0.42**	**0.00337**	**0.81598**

Table 15: Cityscapes diagnostics on separately trained trajectories. The high-variation subset is selected within each run.

	All checkpoints		High variation	
Quantity	FairGrad	APT	FairGrad	APT
Gradient variation	1.669	2.028	2.750	2.967
Error injection	1.352	0.379	2.228	0.272
Tracking error	0.864	0.287	0.955	0.186
Pareto residual	0.161	0.138	0.167	0.189
Update-violation rate	1.34%	0.00%	4.08%	0.00%
Worst-task cosine	0.0399	0.0918	0.0241	0.0657
Task-projection CV	0.558	0.484	0.718	0.614

Independently trained trajectories. The next analysis uses FairGrad and APT-FairGrad with a 74,400-step training budget and 372 Cityscapes checkpoints. "High variation" denotes each trajectory's own top 10% gradient-variation checkpoints, rather than an identical checkpoint subset across the two independently trained runs. The original numerical summaries for gradient variation, error injection, tracking error, and Pareto residual are retained below; their scale should not be compared across different diagnostics. The high-variation Pareto residual is a negative result and is included explicitly. The full numerical comparison is reported in Table 15.

Replay on the same aggregates. To remove the difference in encountered aggregate sequences, fixed-$\beta = 0.9$ momentum is replayed on the APT sequence. This yields the following reported tracking errors. Unlike separate training, replay compares optimizer states at a common sequence of gradient inputs; it does not simulate the counterfactual parameter trajectory that fixed-momentum training would generate.

The initial final-update fraction (Table 16), the independently trained checkpoint-level update-violation rate (Table 15), and the same-sequence hidden-event rate (Table 5) come from different diagnostics. In particular, 1.67%, 1.34%, and 1.88% should not be substituted for one another. Their comparisons support compatible qualitative observations, but their event definitions and samples differ.

Conflict-adjusted association. With C_t defined by Eq. (31), the Cityscapes same-sequence analysis fits

$$B_t^{\text{worst}} = a + b_C C_t + b_V \|G_t - G_{t-1}\| + e_t, \tag{33}$$

where B_t^{worst} is the difference in worst-task update cosine between APT and fixed-momentum replay. The reported standardized conflict coefficient is $b_C = 0.00260$, with 95% CI $[0.00011, 0.00470]$. This supports an association beyond temporal gradient variation, while leaving potential confounding and the relationship to final task metrics unresolved. The paired-bootstrap hidden-event intervals characterize the reported replay comparison; they do not establish an intervention effect on generalization.

Table 16: Initial final-update diagnostic with FairGrad. C_u is the fraction of violated task–update comparisons. Values are reported summaries.

Method	$C_u \downarrow$	Mean cosine↑	Mean β_1
FairGrad	0.0167	0.0731	0.9000
APT-FairGrad	0.0000	0.1419	0.4989

Table 17: Tracking error on the same APT-generated Cityscapes gradient sequence.

Subset	APT	Fixed-β replay	Reduction
Overall	0.287	0.876	67.2%
Top 10% variation	0.186	0.930	80.0%
Final 20%	0.288	0.875	67.1%

H.3 Adaptation mechanisms

Table 18: Adaptation signals and adapted quantities. P is the shared-gradient dimension and K is the number of tasks. Empirical comparisons appear in Table 6.

Method	Signal	Adapted quantity
DEMON	Predefined training-progress schedule	Momentum coefficient
AMM	Two-plane model of the current gradient and historical memory	Momentum coefficient
YellowFin	Curvature range, gradient variance, and distance estimates	Momentum and learning rate
MoCoGrad	Pairwise task conflicts and task-wise momentum	Individual task gradients
APT	Consecutive MTL-aggregate alignment	Momentum coefficient

APT stores one previous aggregate and performs $O(P)$ extra work per step. The reported comparison characterizes DEMON as constant-cost scalar scheduling, AMM as $O(P)$ vector work with constant additional persistent state, YellowFin as $O(P)$ work with $O(P + w)$ state for a window of size w, and MoCoGrad as $O(K^2P)$ pairwise work with $O(KP)$ state. These are conceptual additional costs, not substitutes for implementation-specific timing. The relative result for AMM is competitive; the comparison should not be framed as showing that all alternative adaptive momentum rules fail.

H.4 Measured wall-clock time and memory

Table 19: Reported memory (GB) and complete training time (hours) for FairGrad and APT-FairGrad.

Method	Cityscapes	NYUv2	CelebA
FairGrad	8.578 / 5.75	10.328 / 10.33	5.282 / 2.76
APT-FairGrad	8.794 / 6.95	10.334 / 11.90	5.284 / 2.77
Memory increase	2.52%	0.058%	0.038%
Time increase	20.87%	15.20%	0.36%

APT adds $O(P)$ time and storage for P shared parameters. For FairGrad, training time rises by 20.9%, 15.2%, and 0.4% on Cityscapes, NYUv2, and CelebA, respectively; corresponding memory increases are 0.216, 0.006, and 0.002 GB. Runtime overhead is therefore material on scene understanding. The experiments are carried out on a single NVIDIA A100 GPU. Additional memory is small in the measured cases, whereas the scene-understanding time increases are moderate.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Main contributions are clearly stated in the introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in Section D and the corresponding appendix analyses.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The transition identity and auxiliary tracking, surrogate, and spectral results have explicit assumptions and proofs in Appendices E–G.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [No]

Justification: The main algorithms, benchmark schedules, and available diagnostic protocols are described. Complete reproduction of all additional diagnostics still requires implementation-level settings and analysis details not specified here; no claim of complete disclosure is made.

Guidelines:

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

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

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

   Answer: [No]

   Justification: The benchmark datasets are cited. A public code release with complete reproduction instructions is not documented in this manuscript.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We conduct experiments on the public datasets and follow the mainstream settings.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The original benchmark tables provide three-seed summaries and the replay analysis reports selected paired-bootstrap intervals. Uncertainty is not available for every added ablation or adaptive-baseline comparison; these means are not claimed to establish statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Appendix H.4 reports the GPU types and measured time and memory for selected runs. Per-run device assignments and total research compute, including preliminary experiments, are not fully specified.

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

9. **Code of ethics**

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

   Answer: [Yes]

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper researches on fundamental machine learning problem, which poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have properly credited and respected.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assets are introduced.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We only employ LLM for writing, editing, or formatting purposes.

    Guidelines:

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