MAdam: Metric-Aware Multi-Objective Adam

Fengbei Liu[1] **Rachit Saluja**[1] **Sunwoo Kwak**[1] **Ruibo Wang**[3] **Ruining Deng**[2]
Heejong Kim[2] **Johannes C. Paetzold**[1,2] **Mert R. Sabuncu**[1,2]

[1]Cornell Tech [2]Weill Cornell Medicine [3]TU Delft

Abstract

Multi-objective optimization (MOO) underlies many machine learning problems, yet MOO solvers across the loss-balancing, gradient-balancing, and Pareto-based families almost universally hand their reconciled directions to Adam [22]. We show this coupling introduces two systematic gaps between the solver's intent and the optimizer's execution. The first is a *weighting mismatch*: Adam's second-moment denominator entangles the time-varying preference vector with gradient statistics, marginalizing the preference into a history average and collapsing distinct Pareto trade-offs toward a near-uniform mixture. The second is a *geometric mismatch*: Adam's adaptive metric distorts the Euclidean geometry MOO solvers assume, turning aligned objectives into apparent conflicts. To resolve both jointly, we introduce **MAdam** (Metric-Aware Multi-Objective Adam), a drop-in wrapper that leaves both solver and optimizer unchanged. MAdam preconditions the reconciled direction by the preference-conditioned curvature of the scalarized objective; on this whitened input, Adam's second moment collapses to identity, so the realized update is governed by the preference-conditioned metric. Across multi-task learning, Pareto-front recovery, physics-informed neural networks, and medical imaging, MAdam improves over Adam in aggregate for every solver family. Our code is available at https://github.com/lfb-1/MAdam.

1 Introduction

Many practical learning problems require a shared model to satisfy multiple objectives simultaneously. Such settings are naturally cast as multi-objective optimization (MOO). Three MOO solver paradigms have emerged [5]: *loss-balancing* methods adapt objective weights to rebalance gradient magnitudes [46, 41, 32, 30]; *gradient-balancing* methods modify objective gradients to reduce inter-objective conflict [60, 31, 34]; and *Pareto-based* methods steer the update toward a Pareto front target region [29, 45, 10, 4]. Most gradient based MOO solvers aim to produce a single reconciled direction.

However, gradient-based optimization relies on two steps: *constructing* the descent direction and *executing* the update with a suitable metric, which is a role all three MOO solver families above unanimously delegate to Adam [22]. Adam implicitly imposes its own adaptive diagonal metric via the second-moment denominator, which under stationarity approximates a diagonal empirical Fisher information matrix [23, 39]. We show this default pairing introduces two systematic mismatches between what the solver intends and what the optimizer actually executes.

Firstly, in loss-balancing and Pareto-based methods, the solver's reconciled direction depends on a time-varying preference vector $\boldsymbol{\lambda}^{(t)}$: loss-balancing methods adapt these preferences to rebalance gradient magnitudes, while Pareto-based methods adjust them to steer toward a target Pareto region. However, Adam's running second moment entangles these preferences with the objective gradient statistics into a single surrogate, weakening the intended preference strategy and collapsing distinct Pareto trade-offs into a near-uniform average (Prop. 1).

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

Secondly, MOO solvers typically assume a raw Euclidean geometry: loss-balancing methods explicitly weight the objectives through linear scalarization. Gradient-balancing methods measure gradient similarities and project to reduce conflict in Euclidean space. Pareto-based methods steer a Pareto target region through linear scalarization of user preferences. However, Adam is not metric-neutral: its second-moment denominator imposes a diagonal RMS metric on every update, conflicting with the Euclidean metric assumed by MOO solvers. This geometric mismatch distorts the solver's reconciled direction, turning aligned objectives into apparent conflicts and vice versa (Prop. 2).

These observations raise two questions: **Q1. Which metric should govern MOO updates? Q2. How can it be realized within the existing solver–Adam pipeline?**

We propose MAdam (Metric-aware Multi-objective Adam), which addresses both questions with a single mechanism. For Q1, we derive the *preference-conditioned* diagonal Fisher information matrix of the scalarized objective at the active preference, given by the second moment of the scalarized gradient and admitting a decomposition into within- and cross-objective Fisher blocks. For Q2, we precondition the solver's reconciled direction by this diagonal metric prior to the Adam update, modifying neither the solver nor the optimizer. Applying the correction at Adam's input keeps its first- and second-moment EMAs mutually consistent, so the realized parameter update is governed by the preference-conditioned metric. Our contributions are threefold

- **Diagnosis of the solver–Adam mismatch.** We prove two failure modes of solver–Adam coupling in MOO: a *weighting mismatch* that marginalizes the preference (Prop. 1) and a *geometric mismatch* that distorts the MOO solvers' Euclidean geometry (Prop. 2).
- **Metric-aware gradient preconditioning.** We derive the preference-conditioned diagonal Fisher of the scalarized objective and apply it as a plug-and-play preconditioner on the reconciled direction, resolving both mismatches with a single mechanism.

We validate MAdam across diverse MOO solvers and observe consistent improvements over Adam on multi-task learning, Pareto-front recovery, PINN benchmarks, and medical imaging applications.

2 Related Work

Multi-Task and Pareto MTL. Multi-task learning reconciles competing objectives by rescaling per-objective losses, manipulating task gradients, or targeting a preference-conditioned point on the Pareto front. Representative methods range from fixed or learned scalarizations (linear scalarization, uncertainty weighting [21]) through dynamically adapted weights (DWA [35], IMTL [34], CAGrad [31]) to preference-conditioned models that recover the full Pareto front from a single network (PaMaL [9], PaLoRA [10]). Across this family the constructed direction is ultimately consumed by Adam, whose diagonal second-moment EMA entangles the time-varying preference $\boldsymbol{\lambda}^{(t)}$ with per-objective gradient statistics and imposes a task-agnostic RMS metric on each step.

PINN and MIA Loss Balancing. Physics-informed neural networks (PINNs) and medical-image analysis (MIA) exhibit a closely related imbalance: residual, boundary, and reconstruction losses span widely different scales, so a single objective dominates absent explicit rebalancing. The PINNacle benchmark [15] consolidates remedies along three axes, loss reweighting (LRA [53], NTK [54]), residual-based sampling (RAR [56]), and adaptive activations (LAAF/GAAF [19]); in MIA, vanilla Adam [42] and random loss weighting [27] remain the prevailing choices. Such first-order remedies adjust the descent direction yet leave the Adam optimizer's step metric untouched.

Curvature-Aware Optimizers. A separate line of curvature-aware optimizers, spanning natural gradient [1, 39], K-FAC [40], Shampoo [14], AdaHessian [59], Sophia [33], and FAdam [18], targets a single scalar objective and supplants the first-order optimizer wholesale. None preconditions the reconciled direction of a multi-objective solver under a time-varying preference. MAdam occupies this gap: it derives a preference-conditioned diagonal Fisher of the scalarized objective and inserts it into the solver-then-Adam pipeline as a wrapper, leaving both the MOO solver and the optimizer intact. A comprehensive listing of MTL, PINN, MIA, and curvature-aware methods is in App. A.

3 Method

Sec. 3.1 defines preliminaries about MOO solver and Adam. Sec. 3.2 identifies two problems Adam has on reconciled MOO directions: a weight mismatch and a geometric mismatch. Sec. 3.3 derives the correct preference-conditioned curvature (Q1). Sec. 3.4 applies it by preconditioning the solver's output before passing it to Adam (Q2). Sec. 3.5 describes the practical implementation, and Algorithm 1 states the full procedure.

3.1 Preliminaries

MOO Solver Setup. We consider C objectives with per-objective losses $\{\ell_i(\boldsymbol{\theta})\}_{i=1}^{C}$ under shared model parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. In many scenarios, learning can be formulated as: $\boldsymbol{\theta}^*(\boldsymbol{\lambda}^{(t)}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{C} \lambda_i^{(t)} \ell_i(\boldsymbol{\theta})$, where $\boldsymbol{\lambda}^{(t)} \in \Delta^{C-1}$ is a per-step preference vector on the probability simplex. At iteration t, let $\mathbf{g}_i^{(t)} := \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}^{(t)})$ denote the per-objective gradient.

Definition 1 (MOO Solver)**.** Assume a *MOO solver* produces a reconciled update direction through linear scalarization:

$$\mathbf{d}^{(t)} = \sum_{i=1}^{C} \lambda_i^{(t)} \mathbf{g}_i^{(t)} = \nabla_{\boldsymbol{\theta}} \ell_{\boldsymbol{\lambda}^{(t)}}\big(\boldsymbol{\theta}^{(t)}\big), \qquad \ell_{\boldsymbol{\lambda}^{(t)}} := \sum_{i=1}^{C} \lambda_i^{(t)} \ell_i. \tag{1}$$

The definition requires only a per-step linear combination of per-objective gradients whose coefficients $\boldsymbol{\lambda}^{(t)}$ are detached from $\boldsymbol{\theta}$ in the backward pass; the second equality then follows from linearity of the gradient. It does not require a global scalarized objective or simplex-valued coefficients: $\boldsymbol{\lambda}^{(t)}$ may be signed or unnormalized, as in IMTL, CAGrad, and Nash-MTL. This form unifies three families of MOO solvers, distinguished by how $\boldsymbol{\lambda}^{(t)}$ is specified: (i) *loss balancing*, where $\boldsymbol{\lambda}^{(t)} \in \Delta^{C-1}$ is computed from the latest per-objective losses; (ii) *gradient balancing*, where $\boldsymbol{\lambda}^{(t)} \in \mathbb{R}^C$ is obtained from the geometry of the per-objective gradients (e.g., MGDA, PCGrad, CAGrad); and (iii) *preference-based Pareto*, where $\boldsymbol{\lambda}^{(t)}$ is an externally supplied preference vector that is swept to steer the iterate along the Pareto front. In all three cases, the reconciled direction $\mathbf{d}^{(t)}$ is the *Euclidean gradient* of the scalarized loss $\ell_{\boldsymbol{\lambda}^{(t)}}$ on $\mathbb{R}^d$. Specific solvers in each family are detailed in App. B.

Adam as a Diagonal RMS Preconditioner. Adam [22] maintains exponential moving averages (EMAs) of the first and second moments of the supplied direction $\mathbf{d}^{(t)}$:

$$\mathbf{m}^{(t)} = \beta_1 \mathbf{m}^{(t-1)} + (1-\beta_1)\mathbf{d}^{(t)}, \qquad \mathbf{v}^{(t)} = \beta_2 \mathbf{v}^{(t-1)} + (1-\beta_2)\, \mathbf{d}^{(t)} \odot \mathbf{d}^{(t)}, \tag{2}$$

where $\odot$ denotes element-wise multiplication, and applies the bias-corrected $\hat{\mathbf{m}}^{(t)}$ and $\hat{\mathbf{v}}^{(t)}$:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{\hat{\mathbf{m}}^{(t)}}{\sqrt{\hat{\mathbf{v}}^{(t)}} + \varepsilon}, \tag{3}$$

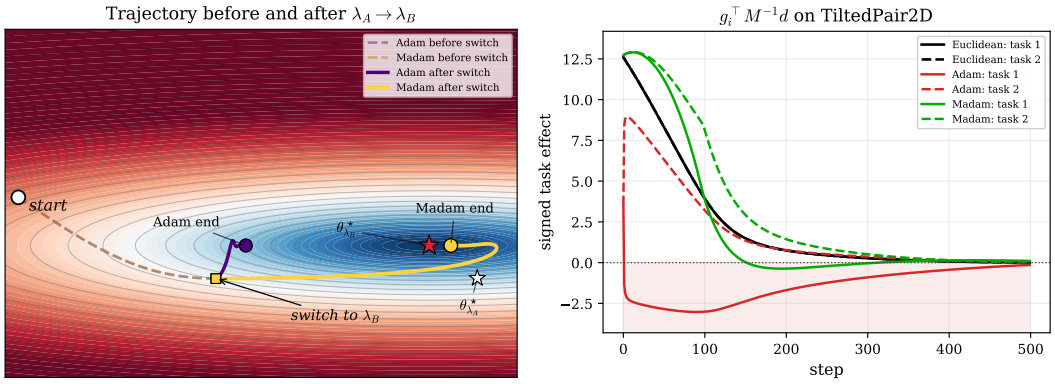
where ε is a small positive constant. Eq. (3) can equivalently be written in preconditioned form as

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \left(\mathbf{M}_{\text{Adam}}^{(t)}\right)^{-1} \hat{\mathbf{m}}^{(t)}, \qquad \mathbf{M}_{\text{Adam}}^{(t)} := \text{Diag}\big(\sqrt{\hat{\mathbf{v}}^{(t)}} + \varepsilon\big), \tag{4}$$

with $\text{Diag}(\cdot)$ the diagonal matrix formed from its vector argument. In a stationary regime, $\hat{\mathbf{v}}^{(t)} \approx \mathbb{E}[\mathbf{d}^{(t)} \odot \mathbf{d}^{(t)}]$ recovers the diagonal of the empirical Fisher information matrix of the update direction [23, 39], so that $\big(\mathbf{M}_{\text{Adam}}^{(t)}\big)^2 \approx \text{Diag}\big(\mathbf{F}^{(t)}\big)$. Eq. (4) therefore realizes approximate natural gradient descent whose metric is a diagonal empirical Fisher information matrix [1, 39, 18];

3.2 Adam and Its Problems in MOO

Source 1: Weighting Mismatch. Adam preconditions its input as though the preference were stationary. When the MOO solver supplies a non-stationary $\boldsymbol{\lambda}^{(t)}$, as in loss-balancing and Pareto-based methods, Adam's second-moment EMA conflates preference weights with gradient statistics and absorbs the current preference into a history-averaged quantity rather than applying it at each step. Fig. 1 (left) illustrates the consequence: after a preference switch $\boldsymbol{\lambda}_A \to \boldsymbol{\lambda}_B$, Adam's iterate remains biased toward the previous optimum $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}_A}$, whereas MAdam converges to $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}_B}$.



Weighting Mismatch **Geometric Mismatch**

Figure 1: **Two sources of solver–Adam mismatch on a 2D tilted-quadratic MOO problem.** *Left (weighting mismatch):* decision-space trajectories across a single preference switch $\boldsymbol{\lambda}_A \to \boldsymbol{\lambda}_B$; contours show the post-switch loss $\ell_{\boldsymbol{\lambda}_B}$ (darker is lower) and the star marks its minimum $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}_B}$. Adam's second-moment EMA still encodes the pre-switch $\boldsymbol{\lambda}_A$, so its iterate stalls near $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}_A}$, while MAdam re-conditions to the active preference and reaches $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}_B}$ (Prop. 1; multi-phase version in App. D.1). *Right (geometric mismatch):* per-objective signed effect $\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d}$ on the same problem. Under the Euclidean metric ($\mathbf{M} = \mathbf{I}$) it is non-negative for every task, so any negative value is a sign flip against the solver's intent. Adam's preconditioner drives task 1 strongly negative (shaded), so the update increases ℓ_1 although the solver treats task 1 as aligned with $\mathbf{d}$; MAdam tracks the Euclidean reference and largely removes the flip (Prop. 2; mechanism in App. C).

Proposition 1 (Adam Tracks Weight-Marginalized Curvature)**.** Assume $\boldsymbol{\lambda}^{(\tau)}$ is independent of the per-objective gradient cross-moments $\mathbf{g}_i^{(\tau)} \odot \mathbf{g}_j^{(\tau)}$ within the EMA window. This assumption holds for loss-balancing solvers, where $\boldsymbol{\lambda}^{(\tau)}$ is determined by scalar loss statistics that vary on a slower timescale than gradient noise, and for preference-based Pareto methods, where $\boldsymbol{\lambda}^{(\tau)}$ is supplied externally. Under this independence and the stationarity approximation in Eq. (4), Adam's second-moment EMA expands as

$$\hat{\mathbf{v}}^{(t)} \;\approx\; \mathbb{E}\big[\mathbf{d}^{(t)} \odot \mathbf{d}^{(t)}\big] \;=\; \sum_{i,j=1}^{C} \mathbb{E}\big[\lambda_i \lambda_j\, \mathbf{g}_i \odot \mathbf{g}_j\big] \;=\; \sum_{i,j=1}^{C} \mathbb{E}[\lambda_i \lambda_j]\, \mathbf{F}_{ij}. \tag{5}$$

The middle expression in (5) makes the entanglement explicit: each summand is a single expectation of $\lambda_i \lambda_j\, \mathbf{g}_i \odot \mathbf{g}_j$ taken jointly. Adam cannot factor this expectation into separate preference and gradient EMAs, so the current weights $\lambda_i^{(t)} \lambda_j^{(t)}$ are subsumed into the history-averaged statistic $\mathbb{E}[\lambda_i \lambda_j]$, which we refer to as the *weight marginalization*. In loss-balancing solvers [46, 41, 32, 30], each $\boldsymbol{\lambda}^{(t)}$ encodes a deliberate re-weighting that Adam's EMA absorbs into a history-averaged $\mathbb{E}[\lambda_i \lambda_j]$, dampening the strategy in the actual update. In preference-based Pareto methods [29, 45, 10, 4] that sweep $\boldsymbol{\lambda}^{(t)}$ within a single run to approximate Pareto front, the marginalization collapses the executed geometry toward a near-uniform mixture, lowering the diversity of Pareto trade-offs.

Source 2: Geometric Mismatch. We now turn to general linear scalarization. The linear scalarization in Definition 1 treats $\mathbf{d}^{(t)}$ as a direction under the Euclidean metric. In practice, however, when Adam is used as the optimizer, the step is executed under Adam's diagonal RMS metric $\mathbf{M}_{\text{Adam}}^{(t)}$ (4), which differs from the Euclidean identity. We now quantify how this discrepancy distorts the per-objective loss change realized by each step.

Proposition 2 (Per-objective Loss Change with Adam)**.** Let $\mathbf{d}^{(t)}$ be the reconciled direction supplied to Adam and $\hat{\mathbf{m}}^{(t)}$ Adam's bias-corrected first-moment EMA of its input. Under the first-order Taylor approximation $\Delta\ell_i^{(t)} \approx \mathbf{g}_i^{(t)\top} \Delta\boldsymbol{\theta}^{(t)}$, the per-objective loss change realized by Adam is

$$\Delta\ell_i^{(t)} \approx -\eta\, \mathbf{g}_i^{(t)\top} \big(\mathbf{M}_{\text{Adam}}^{(t)}\big)^{-1} \hat{\mathbf{m}}^{(t)}. \tag{6}$$

Under the stationarity approximation $\hat{\mathbf{m}}^{(t)} \approx \mathbf{d}^{(t)}$, this reduces to $-\eta\, \mathbf{g}_i^{(t)\top} (\mathbf{M}_{\text{Adam}}^{(t)})^{-1} \mathbf{d}^{(t)}$.

Proof. Recall from (4) that $\mathbf{M}_{\text{Adam}}^{(t)} = \text{Diag}(\sqrt{\hat{\mathbf{v}}^{(t)}} + \varepsilon)$ is the diagonal RMS metric: its inverse is the inverse-square-root-of-empirical-Fisher preconditioner Adam applies coordinate-wise. By the first-order Taylor approximation,

$$\Delta \ell_i^{(t)} \approx \mathbf{g}_i^{(t)\top} \Delta \boldsymbol{\theta}^{(t)}. \tag{7}$$

From (3) and (4), the parameter step is

$$\Delta \boldsymbol{\theta}^{(t)} = -\eta \, \frac{\hat{\mathbf{m}}^{(t)}}{\sqrt{\hat{\mathbf{v}}^{(t)}} + \varepsilon} = -\eta \left(\mathbf{M}_{\text{Adam}}^{(t)}\right)^{-1} \hat{\mathbf{m}}^{(t)}. \tag{8}$$

Substituting into the Taylor expansion gives (6):

$$\Delta \ell_i^{(t)} \approx \mathbf{g}_i^{(t)\top} \left(-\eta \left(\mathbf{M}_{\text{Adam}}^{(t)}\right)^{-1} \hat{\mathbf{m}}^{(t)}\right) = -\eta\, \mathbf{g}_i^{(t)\top} \left(\mathbf{M}_{\text{Adam}}^{(t)}\right)^{-1} \hat{\mathbf{m}}^{(t)}. \tag{9}$$

The stationarity approximation $\hat{\mathbf{m}}^{(t)} \approx \mathbf{d}^{(t)}$ (stated in the proposition) is used in the subsequent analysis of §3.2 and §3.4. □

Eq. (6) expresses the per-objective loss change as a bilinear form in the per-objective gradient $\mathbf{g}_i^{(t)}$ and the first-moment direction $\hat{\mathbf{m}}^{(t)}$ (which coincides with the reconciled $\mathbf{d}^{(t)}$ under stationarity), preconditioned by $(\mathbf{M}_{\text{Adam}}^{(t)})^{-1}$. For $C = 1$, $\mathbf{g}_i^{(t)} = \mathbf{d}^{(t)}$ and the form collapses to the quadratic $-\eta\, \mathbf{d}^{(t)\top} (\mathbf{M}_{\text{Adam}}^{(t)})^{-1} \mathbf{d}^{(t)} \leq 0$, guaranteeing descent for any positive-definite metric. For $C > 1$, however, $\mathbf{g}_i^{(t)} \neq \mathbf{d}^{(t)}$ in general and the bilinear form is indefinite: $(\mathbf{M}_{\text{Adam}}^{(t)})^{-1}$ can flip the sign of $\mathbf{g}_i^{(t)\top} (\mathbf{M}_{\text{Adam}}^{(t)})^{-1} \mathbf{d}^{(t)}$ relative to $\mathbf{g}_i^{(t)\top} \mathbf{d}^{(t)}$, converting an aligned objective into an apparent conflict (App. C). Fig. 1 (right) instantiates this regime on a synthetic two-task problem: Adam's preconditioner drives task 1's signed effect $\mathbf{g}_1^{(t)\top} (\mathbf{M}_{\text{Adam}}^{(t)})^{-1} \mathbf{d}^{(t)}$ strongly negative, producing a large positive $\Delta \ell_1^{(t)}$: the update actively increases the loss the solver intended to decrease.

3.3 Q1: The Correct Curvature for MOO

As we have discussed above, Adam's EMA averages the entangled product of $\boldsymbol{\lambda}^{(t)}$ with the per-objective gradients inside $\mathbf{d}^{(t)}$, rather than the two factors separately. Our proposed remedy is to accumulate gradient cross-moments independently and combine them with the current preference at each step, instead of letting a joint EMA absorb the mixing. The resulting curvature estimate is

$$\mathbf{C}_{\boldsymbol{\lambda}^{(t)}} = \mathbb{E}[\mathbf{g}_{\boldsymbol{\lambda}^{(t)}} \odot \mathbf{g}_{\boldsymbol{\lambda}^{(t)}}] = \sum_{i,j=1}^{C} \lambda_i^{(t)} \lambda_j^{(t)}\, \mathbf{F}_{ij} = \sum_{i=1}^{C} \left(\lambda_i^{(t)}\right)^2 \mathbf{F}_{ii} \;+\; \sum_{i \neq j} \lambda_i^{(t)} \lambda_j^{(t)}\, \mathbf{F}_{ij}, \tag{10}$$

the diagonal second moment of the scalarized gradient at the current preference $\boldsymbol{\lambda}^{(t)}$. The expectation parallels Adam's $\mathbb{E}[\mathbf{d}^{(t)} \odot \mathbf{d}^{(t)}]$ in (5) but holds $\boldsymbol{\lambda}^{(t)}$ fixed at the current step, so the deterministic weights factor out and only the gradient cross-moments remain inside the expectation. The second line separates the diagonal terms $\mathbf{F}_{ii}$ (per-objective curvature) from the off-diagonal terms $\mathbf{F}_{ij}$ (cross-objective gradient correlation).

Two properties confirm $\mathbf{C}_{\boldsymbol{\lambda}^{(t)}}$ as the appropriate metric for the current trade-off. **(i) Current-preference alignment.** The weights $\lambda_i^{(t)} \lambda_j^{(t)}$ appear outside the expectation and $\mathbf{F}_{ij}$ inside, so the current preference directly controls the contribution of each cross-moment rather than being smoothed into a window average. **(ii) Cross-objective structure.** The off-diagonal terms $\mathbf{F}_{ij}$ ($i \neq j$) capture coordinate-wise gradient correlation between tasks: positive entries indicate coordinates where tasks cooperate, negative entries indicate conflict. Adam collapses both diagonal and off-diagonal contributions into a single per-coordinate magnitude; $\mathbf{C}_{\boldsymbol{\lambda}^{(t)}}$ keeps them separable and adjustable as $\boldsymbol{\lambda}^{(t)}$ changes. We define the corresponding preference-conditioned metric as

$$\mathbf{M}_{\boldsymbol{\lambda}^{(t)}} := \text{Diag}\big(\sqrt{\mathbf{C}_{\boldsymbol{\lambda}^{(t)}}} + \varepsilon\big). \tag{11}$$

3.4 Q2: Metric Preconditioning of the Solver Output

Q1 supplies the target metric $\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}$. Eq. (6) offers three intervention points: (i) modify $\mathbf{g}_i^{(t)}$, but each MOO solver handles per-objective gradients differently, requiring a method-specific modification for each solver; (ii) replace $\mathbf{M}_{\text{Adam}}^{(t)}$ with $\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}$ in Adam's denominator, but this adjusts only the second-moment EMA while the first-moment EMA remains built from unscaled inputs, leaving the numerator and denominator inconsistent; (iii) precondition the reconciled direction $\mathbf{d}^{(t)}$, the single output common to all MOO solvers, which propagates the preference correction into both EMAs simultaneously without modifying the solver or the optimizer.

We adopt approach (iii) and supply Adam with the preconditioned direction $\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)}$ in place of $\mathbf{d}^{(t)}$. By construction $\mathbf{M}_{\boldsymbol{\lambda}}$ whitens the scalarized gradient, so Adam's second-moment EMA on this input collapses to identity:

$$\hat{\mathbf{v}}^{(t)} \;\approx\; \mathbb{E}\big[\mathbf{M}_{\boldsymbol{\lambda}}^{-1}\mathbf{d}^{(t)} \odot \mathbf{M}_{\boldsymbol{\lambda}}^{-1}\mathbf{d}^{(t)}\big] \;=\; \mathbf{M}_{\boldsymbol{\lambda}}^{-2}\,\mathbf{C}_{\boldsymbol{\lambda}} \;\approx\; \mathbf{1}, \qquad \mathbf{M}_{\text{Adam}}^{(t)} \;\approx\; \mathbf{I}, \tag{12}$$

making Adam's denominator a pass-through. With the preconditioned input $\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)}$ fed to Adam, stationarity gives $\hat{\mathbf{m}}^{(t)} \approx \mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)}$ and (12) gives $\mathbf{M}_{\text{Adam}}^{(t)} \approx \mathbf{I}$; substituting both into (6):

$$\Delta\ell_i^{(t)} \;\approx\; -\eta\,\mathbf{g}_i^{(t)\top}\big(\mathbf{M}_{\text{Adam}}^{(t)}\big)^{-1}\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)} \;\approx\; -\eta\,\mathbf{g}_i^{(t)\top}\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)}. \tag{13}$$

The reduction of $\mathbf{M}_{\text{Adam}}^{(t)}$ to the identity does not recover the Euclidean metric: it replaces an uncontrolled, history-marginalized preconditioner with the preference-conditioned $\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}$, which is precisely the curvature established in Q1. The realized parameter update is therefore

$$\boldsymbol{\theta}^{(t+1)} \;=\; \boldsymbol{\theta}^{(t)} - \eta\,\mathbf{M}_{\boldsymbol{\lambda}^{(t)}}^{-1}\mathbf{d}^{(t)}. \tag{14}$$

3.5 Practical Estimation

Cross-Objective Fisher Estimation. The metric $\mathbf{M}_{\boldsymbol{\lambda}}$ requires the cross-objective Fisher interactions $\mathbf{F}_{ij}$, which are not directly available. We estimate them online with EMAs that share the same decay rate β_2 as Adam, then assemble and rampup-blend the resulting curvature into the preconditioner:

$$\widehat{\mathbf{F}}_{ij}^{(t)} = \beta_2\,\widehat{\mathbf{F}}_{ij}^{(t-1)} + (1-\beta_2)\,\mathbf{g}_i^{(t)} \odot \mathbf{g}_j^{(t)}, \qquad 1 \le i \le j \le C, \tag{15}$$

$$\widehat{\mathbf{C}}_{\boldsymbol{\lambda}}^{(t)} := \sum_{i,j=1}^{C} \lambda_i^{(t)}\lambda_j^{(t)}\,\widehat{\mathbf{F}}_{ij}^{(t)}, \qquad \widehat{\mathbf{M}}_{\boldsymbol{\lambda}}^{(t)} := \operatorname{Diag}\Big(\alpha^{(t)}\sqrt{\widehat{\mathbf{C}}_{\boldsymbol{\lambda}}^{(t)} + \varepsilon} + (1-\alpha^{(t)})\Big), \tag{16}$$

where the rampup coefficient $\alpha^{(t)} \in [0,1]$ rises monotonically from 0 to 1 over a fixed warmup window, interpolating the denominator between identity (Adam on $\mathbf{d}^{(t)}$) and the full MAdam preconditioner. The rampup defers reliance on $\widehat{\mathbf{F}}_{ij}^{(t)}$ until its EMA has stabilized, since early estimates are dominated by noise. Replacing Eq. (14) $\mathbf{M}_{\boldsymbol{\lambda}}$ with $\widehat{\mathbf{M}}_{\boldsymbol{\lambda}}^{(t)}$ yields the practical algorithm.

Stochastic Pair Sampling. Maintaining the $C(C+1)/2$ symmetric EMA entries (15) requires the per-objective gradient $\mathbf{g}_i^{(t)}$ for every task, incurring $O(C)$ backward passes per step. We reduce this to a constant by drawing a pair $(i,j) \sim \text{Uniform}([C]\times[C])$ at each iteration, computing only $\mathbf{g}_i^{(t)}$ and $\mathbf{g}_j^{(t)}$, and updating the three EMAs that depend on the pair, namely $\widehat{\mathbf{F}}_{ii}^{(t)}$, $\widehat{\mathbf{F}}_{jj}^{(t)}$, and $\widehat{\mathbf{F}}_{ij}^{(t)}$ (with $\widehat{\mathbf{F}}_{ji}$ identified by symmetry); all other entries are retained from step $t-1$. The MOO solver is unchanged and operates on the full task set, computing $\mathbf{d}^{(t)}$ as in Definition 1. This reduces to $O(1)$ backward passes, twice per step, regardless of C.

4 Experiments

Benchmarks. We evaluate MAdam across four MOO regimes that probe distinct sources of preconditioning mismatch. **Multi-task learning (MTL)** on `MultiMNIST` [25], `SARCOS` [51], `UTKFace` [63], `Cityscapes` [8], and `NYUv2` [48], spanning $C \in [2,7]$ tasks across classification,

Algorithm 1 MAdam: Metric-Aware Multi-Objective Adam

Require: MOO solver, C tasks, initial $\boldsymbol{\theta}^{(0)}$, Adam hyperparameters $\eta, \beta_1, \beta_2, \varepsilon$

Initialize Adam state; $\widehat{\mathbf{F}}_{ij} \leftarrow \mathbf{0}$ for $1 \le i \le j \le C$
for $t = 1, 2, \ldots$ **do**
  Receive preference $\boldsymbol{\lambda}^{(t)} \in \Delta^{C-1}$
  Sample pair $(i, j) \sim \mathrm{Uniform}([C] \times [C])$ and compute $\mathbf{g}_i^{(t)}, \mathbf{g}_j^{(t)}$
  Run MOO solver on the full task set to obtain reconciled direction $\mathbf{d}^{(t)}$
  Update $\widehat{\mathbf{F}}_{ii}^{(t)}, \widehat{\mathbf{F}}_{jj}^{(t)}, \widehat{\mathbf{F}}_{ij}^{(t)}$ via (15) (with $\widehat{\mathbf{F}}_{ji}^{(t)} = \widehat{\mathbf{F}}_{ij}^{(t)}$) *// step 1: Fisher EMAs*
  Form $\widehat{\mathbf{C}}^{(t)}, \widehat{\mathbf{M}}^{(t)}$ via (16) *// step 2: assemble metric*
  $\tilde{\mathbf{d}}^{(t)} \leftarrow \big(\widehat{\mathbf{M}}^{(t)}\big)^{-1} \mathbf{d}^{(t)}$ *// step 3: precondition direction*
  $\boldsymbol{\theta}^{(t+1)} \leftarrow \mathrm{Adam}\big(\tilde{\mathbf{d}}^{(t)}\big)$ // standard Adam on preconditioned direction
end for

regression, dense prediction, and surface-normal estimation. **Pareto MTL** on the same five datasets under preference-conditioned, time-varying $\boldsymbol{\lambda}^{(t)}$, where the goal is to trace the Pareto front rather than reach a single trade-off. **Physics-informed Neural Networks (PINNs)** on the PINNacle benchmark [15] with 20 PDEs whose residual / boundary / initial-condition losses live on widely different scales. **Medical image analysis (MIA)** on `ISIC2018` [7] skin-lesion segmentation, where soft-Dice and pixel-wise cross-entropy form the two objectives, and a super-resolution task on `OASIS3` [24], where a $4\times$ downsampled image is reconstructed to its original resolution using multiple reconstruction objectives. Per-dataset multi-objectives, backbones, losses, and metrics are in App. E.

Baselines. We apply MAdam on top of representative MOO solvers and training recipes from each regime, holding all other settings fixed within a baseline/MAdam block. For both MTL and Pareto MTL we follow PaLoRA's recipe (learning rate, backbone, batch size, schedule, per-objective losses) and compare against the standard MTL solvers linear scalarization (LS), Uncertainty Weighting (UW) [21], Dynamic Weight Average (DWA) [35], Impartial MTL (IMTL) [34], and CAGrad [31] for the single-trade-off setting, and against the Pareto-front solvers PaMaL [9] and PaLoRA [10] for the preference-conditioned setting. UW, DWA, IMTL, and CAGrad were proposed between 2018 and 2021, so on `SARCOS` we additionally wrap three recent solvers under the same seven-task setup, Adam configuration, schedule, and seeds: the gradient-balancing Nash-MTL [41] and FairGrad [2], which determine $\boldsymbol{\lambda}^{(t)}$ from the Gram matrix of per-objective gradients (App. B.1), and the loss-balancing ExcessMTL [17], which sets $\boldsymbol{\lambda}^{(t)}$ from per-task excess risks (App. B.2). All three emit the linear form of Eq. (1) with $\boldsymbol{\theta}$-detached coefficients, so MAdam applies without modification. For PINNs we span PINNacle's three baseline families: Vanilla (PINN), Loss Reweighting/Sampling (LRA [53], NTK [54], RAR [56]), and adaptive activations (LAAF/GAAF [19]). For skin-lesion segmentation we compare MAdam against Adam under UNet [44] and SwinTransformer [36]. For super-resolution we compare against UW and LS under Adam. All experiments are implemented in PyTorch [42].

MTL results MAdam improves the base solver in aggregate on every MTL benchmark (Tabs. 1–3 and App. F.1): it dominates all three task accuracies on `MultiMNIST`, most per-joint errors on `SARCOS`, both segmentation and depth on `Cityscapes` (IMTL relative depth error $74.2 \rightarrow 61.8$), and most metrics on `NYUv2`. The remaining exceptions arise when the base solver already balances tasks well, with gaps within one standard deviation. On the recent solvers (Tab. 2), MAdam lowers the `SARCOS` average error of Nash-MTL, FairGrad, and ExcessMTL and improves 20 of 21 task-wise means; the single exception is t_1 under Nash-MTL, where the two values differ by 0.02 and lie within one standard deviation.

Pareto MTL results On `MultiMNIST` (Fig. 2), `SARCOS` (Tab. 4), `Cityscapes` (Tab. 3), and `NYUv2` (Tab. 6). MAdam raises hypervolume and improves both PaLoRA and PaMaL across all benchmarks: it dominates the base front on `MultiMNIST`, reduces per-ray error on nearly every `SARCOS` ray, improves both segmentation metrics on `Cityscapes` with only marginal depth trade-offs, and yields gains on a majority of `NYUv2` metrics, uniformly on surface normals.

Table 1: `MultiMNIST` and `UTKFace` MTL results (mean ± std over three seeds). Bold marks the better value within each Base/+MAdam pair. `MultiMNIST` rows are top-1 accuracy (↑); `UTKFace` Age is MAE in years (↓), Gender/Race are accuracy (↑).

	LS		DWA [35]		UW [21]		IMTL [34]		CAGrad [31]	
	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam
`MultiMNIST`										
Avg. acc.	94.08±0.37	**95.11±0.32**	94.08±0.34	**95.08±0.25**	94.34±0.31	**95.14±0.44**	93.95±0.29	**95.12±0.25**	94.05±0.24	**95.10±0.35**
TL acc.	94.77±0.34	**95.76±0.38**	94.71±0.36	**95.75±0.31**	95.12±0.46	**95.93±0.51**	94.76±0.26	**95.83±0.20**	94.86±0.23	**95.77±0.28**
BR acc.	93.40±0.75	**94.47±0.27**	93.45±0.74	**94.42±0.19**	93.57±0.86	**94.35±0.38**	93.15±0.68	**94.40±0.33**	93.24±0.71	**94.42±0.42**
`UTKFace`										
Age MAE	9.14±0.38	**8.77±0.25**	9.70±1.03	**8.91±0.19**	8.81±0.42	**8.62±0.39**	8.79±0.66	**8.43±0.14**	**8.28±0.14**	8.61±3.80
Gender Acc	90.76±0.26	**91.08±0.26**	90.58±0.62	**90.83±0.40**	91.14±0.24	**91.45±0.24**	90.94±0.21	**91.05±0.39**	**90.99±0.22**	90.42±0.51
Race Acc	81.60±0.55	**81.92±0.31**	81.27±0.52	**81.52±0.34**	81.22±0.45	**81.29±0.27**	80.75±0.40	80.75±0.60	81.34±0.36	**81.37±2.60**

Table 2: `SARCOS` MTL results (mean ± std over three seeds). Errors are scaled by 100. Bold marks the better value within each Base/+MAdam block. Blocks below the thick rule report recent solvers.

Method	Avg. err. ↓	t_1 ↓	t_2 ↓	t_3 ↓	t_4 ↓	t_5 ↓	t_6 ↓	t_7 ↓
LS	10.99 ± 0.59	1.63 ± 0.07	14.87 ± 0.15	0.65 ± 0.00	0.19 ± 0.00	50.32 ± 4.08	8.96 ± 0.30	**0.30 ± 0.01**
+ MAdam	**10.00 ± 0.07**	**1.46 ± 0.02**	**13.55 ± 0.06**	**0.61 ± 0.01**	**0.19 ± 0.02**	**45.43 ± 0.16**	**8.47 ± 0.35**	0.31 ± 0.02
DWA [35]	11.11 ± 0.65	1.64 ± 0.06	14.85 ± 0.06	0.66 ± 0.01	0.19 ± 0.01	51.15 ± 4.54	8.98 ± 0.10	0.31 ± 0.02
+ MAdam	**10.09 ± 0.13**	**1.47 ± 0.02**	**13.57 ± 0.41**	**0.62 ± 0.02**	**0.19 ± 0.01**	**45.94 ± 0.96**	**8.51 ± 0.35**	**0.30 ± 0.02**
UW [21]	10.83 ± 0.71	1.45 ± 0.02	14.29 ± 0.33	0.61 ± 0.01	0.18 ± 0.00	50.71 ± 4.64	8.27 ± 0.14	0.30 ± 0.01
+ MAdam	**9.92 ± 0.31**	**1.33 ± 0.02**	**13.19 ± 0.38**	**0.59 ± 0.03**	**0.17 ± 0.02**	**46.14 ± 1.44**	**7.73 ± 0.34**	**0.29 ± 0.02**
IMTL [34]	12.10 ± 0.29	0.84 ± 0.02	14.86 ± 0.16	0.39 ± 0.04	**0.08 ± 0.00**	60.69 ± 1.94	7.70 ± 0.19	0.15 ± 0.00
+ MAdam	**11.05 ± 0.39**	**0.81 ± 0.03**	**13.32 ± 0.11**	**0.34 ± 0.01**	0.08 ± 0.01	**55.64 ± 2.86**	**7.02 ± 0.27**	**0.15 ± 0.02**
CAGrad [31]	10.98 ± 0.63	1.16 ± 0.06	14.88 ± 0.53	0.66 ± 0.33	0.15 ± 0.07	50.30 ± 3.63	9.53 ± 0.55	0.18 ± 0.01
+ MAdam	**10.38 ± 0.43**	**1.11 ± 0.03**	**13.79 ± 0.14**	**0.44 ± 0.02**	**0.11 ± 0.04**	**47.92 ± 3.11**	**9.13 ± 0.30**	**0.17 ± 0.00**
Nash-MTL [41]	11.39 ± 0.54	**0.83 ± 0.02**	14.46 ± 1.24	0.40 ± 0.02	0.09 ± 0.01	56.62 ± 2.49	7.14 ± 0.10	0.16 ± 0.02
+ MAdam	**10.99 ± 0.30**	0.85 ± 0.02	**13.38 ± 0.19**	**0.37 ± 0.01**	**0.08 ± 0.01**	**55.24 ± 1.90**	**6.87 ± 0.05**	**0.14 ± 0.00**
FairGrad [2]	11.42 ± 0.16	0.87 ± 0.05	14.08 ± 0.70	0.39 ± 0.04	0.10 ± 0.02	57.07 ± 1.94	7.22 ± 0.18	0.17 ± 0.02
+ MAdam	**11.11 ± 0.29**	**0.82 ± 0.03**	**13.36 ± 0.27**	**0.38 ± 0.03**	**0.07 ± 0.00**	**56.09 ± 3.26**	**6.92 ± 0.19**	**0.14 ± 0.00**
ExcessMTL [17]	11.09 ± 0.61	1.75 ± 0.11	15.60 ± 0.54	0.70 ± 0.05	0.22 ± 0.02	50.15 ± 4.49	8.91 ± 0.32	0.33 ± 0.01
+ MAdam	**10.43 ± 0.46**	**1.65 ± 0.10**	**14.71 ± 0.37**	**0.67 ± 0.02**	**0.20 ± 0.01**	**47.12 ± 3.44**	**8.36 ± 0.26**	**0.31 ± 0.02**

Table 3: `Cityscapes` MTL mean results. Bold marks the better value within each Base/+MAdam pair.

	LS		DWA		UW		IMTL		CAGrad		PaMaL [9]		PaLoRA [10]	
	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam	Base	+MAdam
mIoU ↑	70.12	**70.89**	70.10	**70.86**	70.20	**71.12**	**70.77**	70.49	69.23	**70.71**	70.35	**70.59**	71.11	**71.34**
Pix Acc ↑	91.90	**92.14**	91.89	**92.18**	91.93	**92.26**	**92.12**	92.02	91.61	**91.63**	91.99	**92.17**	92.21	**92.29**
Abs Err ↓	0.0192	**0.0187**	0.0192	**0.0184**	0.0189	**0.0172**	0.0151	**0.0147**	0.0168	**0.0165**	**0.0141**	0.0143	**0.014**	0.0147
Rel Err ↓	124.06	**123.17**	127.66	**120.31**	125.94	**116.10**	74.23	**61.81**	110.14	**101.95**	**54.52**	54.99	**51.27**	52.27

Table 4: `SARCOS` Pareto MTL results (mean ± std over three seeds). Errors are scaled by 100; r_0–r_7 report per-ray average error. Bold marks the better value within each Base/+MAdam block. Per-task per-ray numbers are tabulated in App. F.2 (Tabs. 9, 8).

Method	HV ↑	Mean ↓	r_0 ↓	r_1 ↓	r_2 ↓	r_3 ↓	r_4 ↓	r_5 ↓	r_6 ↓	r_7 ↓
PaMaL	378.2864 ± 9.2692	48.31 ± 17.53	87.86 ± 87.91	41.22 ± 9.52	**34.91 ± 3.77**	37.00 ± 6.33	61.62 ± 40.73	40.42 ± 5.35	55.28 ± 13.61	28.19 ± 0.85
+ Madam	**394.5543 ± 4.1958**	**38.48 ± 10.98**	**32.06 ± 2.72**	**33.05 ± 1.46**	86.15 ± 89.88	**35.31 ± 5.35**	**32.92 ± 0.52**	**30.59 ± 1.23**	**32.72 ± 2.08**	**25.01 ± 0.74**
PaLoRA	431.2050 ± 8.4737	16.62 ± 3.61	16.53 ± 2.20	16.01 ± 2.37	18.31 ± 7.37	15.53 ± 1.67	15.47 ± 2.02	18.86 ± 7.90	18.52 ± 5.83	13.76 ± 1.19
+ Madam	**442.1679 ± 3.4060**	**13.91 ± 1.64**	**14.83 ± 2.40**	**13.96 ± 2.03**	**13.53 ± 2.24**	**12.45 ± 0.92**	**14.05 ± 1.97**	**15.28 ± 1.89**	**14.88 ± 4.39**	**12.31 ± 0.66**

PINN results Tabs. 10–12 (App. F.3) report mean L2RE on PINNacle for three baseline families, each paired with MAdam: Vanilla (PINN); Loss Reweighting/Sampling (LRA, NTK, RAR); and adaptive activations (LAAF, GAAF). MAdam improves the base solver on most PDEs in every family, with the largest gains on geometrically complex domains, chaotic systems, and high-dimensional Poisson, where residual, boundary, and initial-condition losses span disparate scales; gains com-

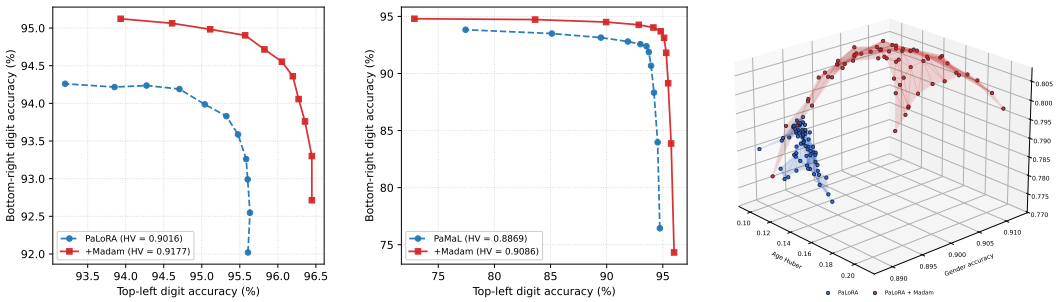


Figure 2: Pareto fronts on `MultiMNIST` (left, center) and `UTKFace` (right). `MultiMNIST` shows the two-task accuracy space (top-left vs. bottom-right digit) under PaLoRA and PaMaL, evaluated on 11 uniformly spaced preference rays. `UTKFace` shows the 3-D metric front under PaLoRA. Per-ray per-objective numbers for `MultiMNIST` are tabulated in App. F.2 (Tab. 7).

`OASIS3` super-resolution metric

Metric	L1 baseline	LS + MAdam
LPIPS ↓	0.2823 ± 0.020	$\mathbf{0.2755 \pm 0.019}$
DISTS ↓	0.2421 ± 0.014	$\mathbf{0.2359 \pm 0.014}$
MAE ↓	0.0425 ± 0.004	$\mathbf{0.0405 \pm 0.004}$
PSNR (dB) ↑	21.244 ± 0.463	$\mathbf{21.522 \pm 0.491}$
SSIM ↑	0.8095 ± 0.019	$\mathbf{0.8181 \pm 0.019}$

`ISIC2018` segmentation: Dice ↑, HD95 ↓

Method	Swin		UNet	
	Dice	HD95	Dice	HD95
LS	0.8849	20.6307	0.8775	21.1575
LS + MAdam	**0.8855**	**20.1855**	0.8772	22.1048
RLW + MAdam	0.8807	20.8601	**0.8821**	**20.9142**

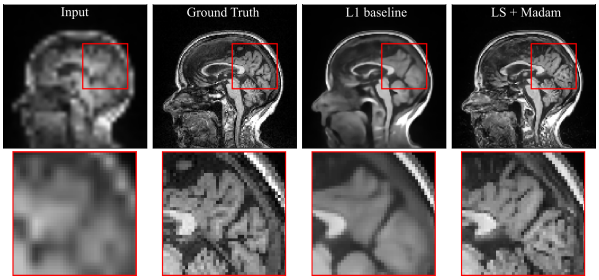


Figure 3: MIA results. **Left top:** `OASIS3` super-resolution validation metrics (mean $\pm$ std over three seeds; LPIPS [61] and DISTS [11] are perceptual, PSNR/SSIM/MAE are pixelwise; bold marks the better value within each Base/+MAdam block). **Left bottom:** `ISIC2018` skin-lesion validation Dice (↑) and HD95 (↓) across backbones; bold marks the better value within each Adam/+MAdam block. **Right:** qualitative `OASIS3` reconstruction results: the zoom-in region outlined in red highlights LS + MAdam better preserves the cortical folds.

pound when MAdam is composed with loss-balancing baselines, while the few cases favoring the base lie in regimes where both saturate near 100% L2RE.

MIA results On `OASIS3` brain-MRI $4\times$ super-resolution (Fig. 3), pairing MAdam with the LS reweighting scheme yields consistent improvements in perceptual fidelity, as measured by LPIPS [61] and DISTS [11], relative to the L1 baseline. These gains extend to the pixel-level metrics (PSNR, SSIM, MAE) as well, so perceptual and pixelwise quality improve jointly; qualitatively, this manifests as sharper preservation of fine cortical structure. On `ISIC2018` segmentation (Fig. 3, bottom-left), MAdam performs on par with Adam under the SwinTransformer backbone, and combining it with RLW [27] under UNet produces the most balanced Dice/HD95 profile among the configurations considered.

Ablation studies We deactivate each of MAdam's three components in turn under LS on `SARCOS` (Tab. 5). Disabling the rampup is most damaging: engaging the preconditioner before per-objective Fisher estimates stabilize drops performance below the LS baseline. Full quadratic enumeration over objective pairs matches the full method at quadratic cost, validating stochastic sampling as an unbiased estimator. Restricting to the diagonal Fisher similarly erodes the gain, indicating off-diagonal curvature between conflicting objectives contributes meaningfully. The full method attains

Table 5: SARCOS ablation under LS scalarization (mean ± std over three seeds). Errors are scaled by 100. Bold marks the best value in each column.

Method	Avg. err. ↓	t_1 ↓	t_2 ↓	t_3 ↓	t_4 ↓	t_5 ↓	t_6 ↓	t_7 ↓
LS + MAdam (full)	**10.00 ± 0.07**	1.46 ± 0.02	**13.55 ± 0.06**	**0.61 ± 0.01**	0.19 ± 0.02	**45.43 ± 0.16**	8.47 ± 0.35	0.31 ± 0.02
w/o FIM rampup Eq. (16)	11.74 ± 0.54	1.76 ± 0.03	16.43 ± 0.97	0.72 ± 0.07	0.22 ± 0.03	52.85 ± 2.27	9.88 ± 0.62	0.34 ± 0.02
w/o stochastic pair sampling	10.33 ± 0.61	**1.45 ± 0.04**	14.02 ± 0.83	0.62 ± 0.03	**0.18 ± 0.01**	47.44 ± 4.53	**8.32 ± 0.25**	**0.30 ± 0.00**
w/o off-diagonal $\mathbf{F}_{ij}$	10.47 ± 0.70	1.50 ± 0.02	14.18 ± 0.35	0.64 ± 0.03	0.18 ± 0.01	48.10 ± 4.93	8.35 ± 0.25	0.30 ± 0.01
LS (baseline)	10.99 ± 0.59	1.63 ± 0.07	14.87 ± 0.15	0.65 ± 0.00	0.19 ± 0.00	50.32 ± 4.09	8.96 ± 0.30	0.30 ± 0.01

the strongest aggregate performance and lowest variance across seeds. App. F.4 further compares MAdam against curvature-aware optimizers (FAdam, Shampoo, AdaHessian) under the same setup.

Runtime and trajectory analysis On SARCOS, MAdam increases per-step wall-clock by 12–19% due to two extra sampled backward passes and $O(C^2)$ Fisher EMA storage, yet CAGrad + MAdam reaches a matched validation error with 20% fewer cumulative FLOPs than CAGrad + Adam (App. G). On a two-dimensional toy problem whose correct local metric rotates with $\boldsymbol{\lambda}$, MAdam follows the rotating stiff direction toward $\mathbf{x}^\star(\boldsymbol{\lambda})$, whereas Adam stays locked to the ambient basis (Fig. 5, App. D.2).

5 Conclusion

We introduced MAdam, a drop-in wrapper that resolves two mismatches in the solver–Adam pipeline: a *weighting mismatch* with Adam's history-averaged second moment, and a *geometric mismatch* with the Euclidean geometry assumed by MOO solvers. Preconditioning the reconciled direction by the preference-conditioned diagonal Fisher lets this metric govern the update. Across multi-task learning, Pareto-front recovery, PINNs, and medical imaging, MAdam consistently improves over Adam for diverse MOO solvers.

Limitations and future work. The main limitation of MAdam is memory: it stores $C(C+1)/2$ cross-objective Fisher EMAs over the shared parameters, so its optimizer state grows quadratically in the number of objectives. Scaling MAdam to many more objectives and to larger models, for example by restricting the stored pairs or sharding the Fisher bank, is left to future work.

Acknowledgments and Disclosure of Funding

This work was partially supported by funding from NewYork-Presbyterian for the NYP-Cornell Cardiovascular AI Collaboration. We also gratefully acknowledge use of the research computing resources of the Empire AI Consortium, Inc, with support from Empire State Development of the State of New York, the Simons Foundation, and the Secunda Family Foundation [3].

References

[1] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[2] H. Ban and K. Ji. Fair resource allocation in multi-task learning. In *International Conference on Machine Learning*. PMLR, 2024.

[3] R. Bloom et al. Empire AI: A multi-institutional shared AI computing infrastructure. In *Proceedings of the Practice and Experience in Advanced Research Computing (PEARC '25)*, 2025. doi: 10.1145/3708035.3736070.

[4] L. Chen, A. Saif, Y. Shen, and T. Chen. Ferero: A flexible framework for preference-guided multi-objective learning. *Advances in Neural Information Processing Systems*, 37:18758–18805, 2024.

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

[6] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

[7] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. 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.

[9] N. Dimitriadis, P. Frossard, and F. Fleuret. Pareto manifold learning: Tackling multiple tasks via ensembles of single-task models. In *International Conference on Machine Learning*. PMLR, 2023.

[10] N. Dimitriadis, P. Frossard, and F. Fleuret. Pareto low-rank adapters: Efficient multi-task learning with preferences. In *International Conference on Learning Representations*, 2025.

[11] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (5):2567–2581, 2022.

[12] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, 2021.

[13] T. George, C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[14] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

[15] Z. Hao, J. Yao, C. Su, H. Su, Z. Wang, F. Lu, Z. Xia, Y. Zhang, S. Liu, L. Lu, et al. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. *Advances in Neural Information Processing Systems*, 37:76721–76774, 2024.

[16] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pages 272–284. Springer, 2021.

[17] Y. He, S. Zhou, G. Zhang, H. Yun, Y. Xu, B. Zeng, T. Chilimbi, and H. Zhao. Robust multi-task learning with excess risks. In *International Conference on Machine Learning*. PMLR, 2024.

[18] D. Hwang. Fadam: Adam is a natural gradient optimizer using diagonal empirical fisher information. *arXiv preprint arXiv:2405.12807*, 2024.

[19] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

[20] A. Javaloy and I. Valera. RotoGrad: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022.

[21] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[23] F. Kunstner, L. Balles, and P. Hennig. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, 2019.

[24] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. G. Vlassenko, et al. Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medrxiv*, pages 2019–12, 2019.

[25] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[26] Y. Li, F. Dangel, D. Tam, and C. Raffel. Fishers for free? approximating the fisher information matrix by recycling the squared gradient accumulator. In *International Conference on Machine Learning*. PMLR, 2025.

[27] B. Lin, F. Ye, Y. Zhang, and I. W. Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022.

[28] B. Lin, W. Jiang, F. Ye, Y. Zhang, P. Chen, Y.-C. Chen, S. Liu, I. W. Tsang, and J. T. Kwok. Dual-balancing for multi-task learning. *Neural Networks*, 195:108317, 2026.

[29] X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[30] X. Lin, X. Zhang, Z. Yang, F. Liu, Z. Wang, and Q. Zhang. Smooth tchebycheff scalarization for multi-objective optimization. *arXiv preprint arXiv:2402.19078*, 2024.

[31] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

[32] B. Liu, Y. Feng, P. Stone, and Q. Liu. Famo: Fast adaptive multitask optimization. In *Advances in Neural Information Processing Systems*, 2023.

[33] H. Liu, Z. Li, D. Hall, P. Liang, and T. Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference on Learning Representations*, 2024.

[34] L. Liu, Y. Li, Z. Kuang, J.-H. Xue, Y. Chen, W. Yang, Q. Liao, and W. Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[35] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[37] D. Mahapatra and V. Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization. In *International Conference on Machine Learning*. PMLR, 2020.

[38] I. Malkiel and L. Wolf. MTAdam: Automatic balancing of multiple training loss terms. In *Conference on Empirical Methods in Natural Language Processing*, pages 10713–10729, 2021.

[39] J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[40] J. Martens and R. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417. PMLR, 2015.

[41] A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, and E. Fetaya. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pages 16428–16446. PMLR, 2022.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[43] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[44] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[45] M. Ruchte and J. Grabocka. Scalable pareto front approximation for deep multi-objective learning. In *IEEE International Conference on Data Mining*, pages 1306–1311. IEEE, 2021.

[46] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[47] D. Senushkin, N. Patakin, A. Kuznetsov, and A. Konushin. Independent component alignment for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[48] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[49] A. Soen and K. Sun. Trade-offs of diagonal fisher information matrix estimators. *Advances in Neural Information Processing Systems*, 37:5870–5912, 2024.

[50] Q. Tong, G. Liang, and J. Bi. Calibrating the adaptive learning rate to improve convergence of ADAM. *Neurocomputing*, 481:333–356, 2022.

[51] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. In *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, volume 1, pages 288–293. Stanford, CA, 2000.

[52] N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, 2025.

[53] S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

[54] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

[55] Z. Wang, Y. Tsvetkov, O. Firat, and Y. Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021.

[56] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

[57] X. Wu, W. Yu, C. Zhang, and P. Woodland. An improved empirical fisher approximation for natural gradient descent. *Advances in Neural Information Processing Systems*, 37:134151–134194, 2024.

[58] E. Yang, J. Pan, X. Wang, H. Yu, L. Shen, X. Chen, L. Xiao, J. Jiang, and G. Guo. AdaTask: A task-aware adaptive learning rate approach to multi-task learning. In *AAAI Conference on Artificial Intelligence*, pages 10745–10753, 2023.

[59] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. W. Mahoney. AdaHessian: An adaptive second order optimizer for machine learning. In *AAAI Conference on Artificial Intelligence*, 2021.

[60] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836, 2020.

[61] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[62] Y. Zhang, C. Chen, T. Ding, Z. Li, R. Sun, and Z.-Q. Luo. Why transformers need Adam: A Hessian perspective. In *Advances in Neural Information Processing Systems*, 2024.

[63] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.

[64] Z. Zhou, L. Cao, P. Wu, P. Zhao, and C. Miao. Delve into the applicability of advanced optimizers for multi-task learning. *arXiv preprint arXiv:2604.08939*, 2026.

Appendix Outline

A Comprehensive Related Work **16**

B Representative MOO Solvers **18**

B.1 Gradient Balancing 18

B.2 Loss Balancing 19

B.3 Preference-Based Pareto 19

B.4 Common Property: Reconciled Directions Are Euclidean 19

C Sign Flips Under Adam's Geometric Mismatch **20**

D Toy-Problem Visualizations **21**

D.1 Multi-Phase Weighting Mismatch 21

D.2 Decision-Space Trajectories Under Rotating Curvature 21

E Experiment Setup: Datasets and Multi-Objectives **24**

E.1 Multi-Task Learning and Pareto MTL Benchmarks 24

E.2 Physics-Informed Neural Networks (PINNs) 25

E.3 Medical Image Analysis 25

F Additional Experimental Results **27**

F.1 `NYUv2` MTL Results 27

F.2 Per-Task Pareto MTL Results 27

F.3 PINN Results 29

F.4 Curvature-Aware Optimizers 30

G Computational Cost **31**

A Comprehensive Related Work

Multi-Task and Pareto MTL. MTL methods reconcile multiple objectives by reshaping per-objective loss weights, projecting or rotating task gradients, or steering toward a preference-conditioned point on the trade-off front, and then hand the resulting direction to a generic optimizer. The baselines we compare against in experiments are Linear Scalarization (LS), the equal-weight sum; Uncertainty Weighting [21], which derives weights from homoscedastic noise; Dynamic Weight Average [35], which reweights by recent loss ratios; IMTL [34], which normalizes loss scale (IMTL-L) and per-objective gradient norms (IMTL-G); CAGrad [31], which maximizes the worst-case task improvement; PaMaL [9], which builds Pareto solutions as preference-weighted ensembles of single-task models; and PaLoRA [10], which extends preference conditioning to low-rank adapters so that varying $\boldsymbol{\lambda}$ at inference traces the entire Pareto front. Beyond these, the broader category includes loss-balancing methods such as GradNorm [6], FAMO [32], Smooth Tchebycheff scalarization [30], DB-MTL [28]. gradient-balancing methods such as PCGrad [60], GradVac [55], Nash-MTL [41], RotoGrad [20], Aligned-MTL [47], GradDrop [49], and TAG [12]; multi-task optimizer variants MTAdam [38] and AdaTask [58]; and additional Pareto-front solvers MGDA [46], PMTL [29], EPO [37], COSMOS [45], and FERERO [4]. Across all of these, the constructed direction is then committed by Adam, whose diagonal second-moment EMA conflates the time-varying preference $\boldsymbol{\lambda}^{(t)}$ with per-objective gradient statistics and imposes a task-agnostic RMS metric on the executed step, so the solver's prescribed weighting and direction are distorted by the metric of the optimizer that every method here defaults to. A concurrent study by Zhou et al. [64] also identifies an incompatibility between MTL-derived directions and adaptive optimizers, but the two works differ in mechanism, optimizer scope, and evaluation. Zhou et al. [64] observe that under adaptive optimizers the instantaneous MTL direction plays only a marginal role in the actual parameter update, because the first-moment EMA lets accumulated momentum dominate; their APT framework therefore adapts the momentum coefficient β_1 so that the current aggregated gradient regains influence. MAdam targets the second moment instead: Adam's second-moment EMA mixes past solver weights into the curvature estimate (Prop. 1) and distorts the Euclidean geometry the solver assumes (Prop. 2), and MAdam reconstructs a preference-conditioned diagonal Fisher from cross-objective moments before the direction reaches Adam. The two corrections act on different EMAs and are in principle composable. In optimizer scope, Zhou et al. [64] study Adam, AdamW, and Muon, and add a separate direction-preservation mechanism for Muon's orthogonalization; MAdam is an Adam wrapper that leaves the optimizer's internal update unchanged and does not modify or evaluate Muon. In evaluation, Zhou et al. [64] evaluate single-solution MTL under a fixed preference; MAdam additionally evaluates preference-conditioned Pareto-front recovery under time-varying $\boldsymbol{\lambda}^{(t)}$, where the weighting mismatch of Prop. 1 is most pronounced, and covers loss-balancing, gradient-balancing, and Pareto-based solvers across MTL, PINNs, and medical imaging.

PINN and MIA Loss Balancing. Physics-informed neural networks (PINNs) and medical-image analysis (MIA) training confront a closely related problem: residual, boundary, and initial-condition losses in PINNs, and perceptual, structural, and pixel-wise reconstruction losses in MIA, live on widely different scales, so a single objective dominates training unless explicitly rebalanced. For PINNs we evaluate against the three PINNacle [15] families: the vanilla PINN [43], which trains a network to satisfy the PDE residual and boundary/initial-condition losses under uniform weighting; LRA [53], which adapts loss weights from gradient-magnitude statistics; NTK [54], which derives per-loss weights from the Neural Tangent Kernel of each loss term; RAR [56], which adaptively resamples residual collocation points by residual magnitude; and LAAF / GAAF [19], which alter expressivity through locally and globally adaptive activation functions rather than weights. For MIA we compare against vanilla Adam [42] on UNet [44] and SwinTransformer [36] backbones, and against RLW [27], which periodically reweights losses by random draws. Other methods surveyed in PINNacle, including causal training, time-marching schedules, and further sampling strategies, share the same first-order spirit and are not separately enumerated. These methods correctly diagnose loss-imbalance and address it with first-order remedies (weights, samples, or activations), but none modifies the geometry under which the rebalanced direction is executed; the optimizer's step metric, supplied by Adam, is left untouched even when the loss landscape it defines is the source of the imbalance.

Curvature-Aware Optimizers. A separate line of work derives parameter updates from second-order or Fisher information rather than the raw gradient, aiming to accelerate convergence on a single scalar objective by replacing the first-order optimizer altogether. None of these methods is used as a baseline in our experiments, but they form the closest neighbor to MAdam on the metric axis. Natural gradient descent [1, 39] preconditions the update by the Fisher information matrix; K-FAC [40] approximates this Fisher in a Kronecker-factored form layer-wise, with EKFAC [13] extending the factorization to its eigenbasis; Shampoo [14] maintains full preconditioners along each tensor axis, and SOAP [52] composes Shampoo's preconditioner with Adam's adaptive scaling; AdaHessian [59] estimates the Hessian diagonal via Hutchinson sampling, and Sophia [33] clips the diagonal Hessian estimate for stable language-model pretraining; FAdam [18] reinterprets Adam itself as a diagonal empirical-Fisher natural-gradient optimizer, and recent diagonal-Fisher estimators [49, 26, 57] sharpen the surrogate Adam already implicitly uses, while Kunstner et al. [23] cautions against treating Adam's denominator as a generic Fisher proxy. All of these methods target a single scalar objective and intervene by replacing the first-order optimizer with a curvature-aware one; none preconditions the reconciled direction of a multi-objective solver under a time-varying preference. MAdam differs in both scope and integration: it derives a preference-conditioned diagonal Fisher of the scalarized objective and slots it into the solver-then-Adam pipeline as a wrapper, leaving both the MOO solver and the optimizer untouched.

B Representative MOO Solvers

This appendix walks through representative gradient-based MOO solvers and verifies for each that the reconciled direction fits the linear-scalarization form of Definition 1,

$$\mathbf{d}^{(t)} \;=\; \sum_{i=1}^{C} \lambda_i^{(t)}\, \mathbf{g}_i^{(t)}, \qquad \mathbf{g}_i^{(t)} \;:=\; \nabla_{\boldsymbol{\theta}} \ell_i\big(\boldsymbol{\theta}^{(t)}\big), \tag{17}$$

with $\boldsymbol{\lambda}^{(t)}$ detached from $\boldsymbol{\theta}$ in the backward pass. Each per-objective gradient $\mathbf{g}_i^{(t)}$ is, by autodiff, the gradient of ℓ_i under the standard Euclidean inner product on $\mathbb{R}^d$. Linearity of the gradient operator with $\boldsymbol{\theta}$-detached coefficients then gives

$$\mathbf{d}^{(t)} \;=\; \nabla_{\boldsymbol{\theta}}\, \ell_{\boldsymbol{\lambda}^{(t)}}\big(\boldsymbol{\theta}^{(t)}\big), \qquad \ell_{\boldsymbol{\lambda}^{(t)}} \;:=\; \sum_{i=1}^{C} \lambda_i^{(t)} \ell_i, \tag{18}$$

the Euclidean gradient of the scalarized loss on $\mathbb{R}^d$. The three families below differ only in how $\boldsymbol{\lambda}^{(t)}$ is determined; the Euclidean property is inherited uniformly, and so is the geometric mismatch with Adam's diagonal RMS metric (Sec. 3.2).

For the gradient-balancing methods we use the per-objective gradient matrix $\mathbf{G}^{(t)} := [\mathbf{g}_1^{(t)}, \dots, \mathbf{g}_C^{(t)}] \in \mathbb{R}^{d\times C}$ and its Gram $\mathbf{K}^{(t)} := \mathbf{G}^{(t)\top}\mathbf{G}^{(t)} \in \mathbb{R}^{C\times C}$ with $[\mathbf{K}^{(t)}]_{ij} = \mathbf{g}_i^{(t)\top}\mathbf{g}_j^{(t)}$. Each Gram entry is the trace of the cross-objective Fisher diagonal $\mathbf{F}_{ij}^{(t)} := \mathbb{E}[\mathbf{g}_i^{(t)} \odot \mathbf{g}_j^{(t)}] \in \mathbb{R}^d$, namely $[\mathbf{K}^{(t)}]_{ij} = \mathbf{1}^\top \mathbf{F}_{ij}^{(t)}$ in expectation; MAdam retains the full per-coordinate $\mathbf{F}_{ij}^{(t)}$ used to assemble the preference-conditioned metric in Eq. (11).

B.1 Gradient Balancing

Gradient-balancing solvers determine $\boldsymbol{\lambda}^{(t)}$ implicitly from the geometry of the per-objective gradients (typically a function of $\mathbf{K}^{(t)}$). The output is a linear combination of $\{\mathbf{g}_i^{(t)}\}$ with coefficients detached from $\boldsymbol{\theta}$.

MGDA [46]. Picks $\boldsymbol{\lambda}^{(t)} \in \Delta^C$ as the minimum-norm element of the convex hull of the per-objective gradients,

$$\boldsymbol{\lambda}^{(t)} \;=\; \arg\min_{\boldsymbol{\lambda}\in\Delta^C} \boldsymbol{\lambda}^\top \mathbf{K}^{(t)} \boldsymbol{\lambda}, \tag{19}$$

solved as a small QP in $\mathbb{R}^C$. The reconciled direction $\mathbf{d}^{(t)} = \sum_i \lambda_i^{(t)} \mathbf{g}_i^{(t)}$ is by construction (17); the QP is solved with stop-gradient on $\mathbf{K}^{(t)}$, so $\boldsymbol{\lambda}^{(t)}$ is detached.

PCGrad [60]. For each pair (i, j) with $\mathbf{g}_i^{(t)\top}\mathbf{g}_j^{(t)} < 0$, PCGrad replaces $\mathbf{g}_i^{(t)}$ by its projection onto the orthogonal complement of $\mathbf{g}_j^{(t)}$ and sums the projected vectors. Each projection is linear in $\{\mathbf{g}_k^{(t)}\}$ with coefficients drawn from $\mathbf{K}^{(t)}$, so the output collects into (17) with effective coefficients

$$\lambda_i^{(t)} \;=\; 1 \;-\; \sum_{j\in\mathcal{C}_i^{(t)}} \frac{\mathbf{g}_j^{(t)\top}\mathbf{g}_i^{(t)}}{\big\|\mathbf{g}_i^{(t)}\big\|^2}, \tag{20}$$

where $\mathcal{C}_i^{(t)}$ indexes objectives that conflict with task i at step t. Coefficients lie in $\mathbb{R}^C$ rather than Δ^C, but the form is still linear scalarization with detached weights.

CAGrad [31]. Solves a trust-region problem around the average gradient $\mathbf{g}_0^{(t)} := \frac{1}{C}\sum_i \mathbf{g}_i^{(t)}$, returning

$$\mathbf{d}^{(t)} \;=\; \mathbf{g}_0^{(t)} + \alpha^* \mathbf{G}^{(t)} \boldsymbol{w}^*, \qquad \boldsymbol{w}^* \in \Delta^C, \tag{21}$$

with $\boldsymbol{w}^*$ from a small QP and trust-region radius α^*. Substituting $\mathbf{g}_0^{(t)} = \mathbf{G}^{(t)}(\mathbf{1}/C)$ shows $\mathbf{d}^{(t)}$ is (17) with effective weights $\lambda_i^{(t)} = 1/C + \alpha^* w_i^*$.

Other gradient-balancing methods. IMTL-G [34] chooses $\boldsymbol{\lambda}^{(t)}$ in closed form so that the reconciled direction has equal projections onto each unit-normalized $\mathbf{g}_i^{(t)}$. RotoGrad [20] rotates per-objective feature spaces so that per-objective gradients align in magnitude, then sums the rotated gradients (taking the rotated $\tilde{\mathbf{g}}_i^{(t)}$ as the per-objective gradients leaves the form unchanged). Aligned-MTL [47] rebalances gradients in the eigenbasis of $\mathbf{K}^{(t)}$, Nash-MTL [41] solves a bargaining QP for $\boldsymbol{\lambda}^{(t)}$, and FairGrad [2] chooses $\boldsymbol{\lambda}^{(t)}$ so that the per-objective improvements $\mathbf{g}_i^{(t)\top}\mathbf{d}^{(t)}$ follow an α-fair allocation, which reduces to a nonlinear system in $\mathbf{K}^{(t)}$ and $\boldsymbol{\lambda}^{(t)}$. All produce (17) with weights determined by $\mathbf{K}^{(t)}$ and detached from $\boldsymbol{\theta}$.

B.2 Loss Balancing

Loss-balancing solvers determine $\boldsymbol{\lambda}^{(t)}$ from the history of scalar losses $\{\ell_i^{(\tau)}\}_{\tau \le t}$ and form $\mathbf{d}^{(t)}$ via (17) with $\boldsymbol{\lambda}^{(t)}$ detached from $\boldsymbol{\theta}$ in the backward pass.

GradNorm [6]. Adjusts $\boldsymbol{\lambda}^{(t)}$ by a separate optimization step that drives per-objective gradient norms toward relative training-rate targets. The main update consumes the resulting $\boldsymbol{\lambda}^{(t)}$ as linear-scalarization weights.

FAMO [32]. Maintains $\boldsymbol{\lambda}^{(t)} = \mathrm{softmax}(\boldsymbol{z}^{(t)})$ and updates the logits $\boldsymbol{z}^{(t)}$ via a closed-form rule based on per-objective log-loss decrease rates. The reconciled direction is again (17).

ExcessMTL [17]. Sets $\boldsymbol{\lambda}^{(t)}$ from per-task excess risks, the gap between the current loss and an estimate of the task's optimal loss, so that tasks farther from convergence receive larger weights. The main update consumes $\boldsymbol{\lambda}^{(t)}$ as linear-scalarization weights.

Uncertainty Weighting [21]. Learns per-objective observation noises σ_i jointly with $\boldsymbol{\theta}$; the effective per-objective weight at the main step is $\lambda_i^{(t)} = 1/\big(2(\sigma_i^{(t)})^2\big)$ (modulo regularization terms in the loss). The detached weight feeds into (17).

B.3 Preference-Based Pareto

Preference-based Pareto methods take $\boldsymbol{\lambda}^{(t)}$ as an external input, typically a user preference on the simplex, and use it to steer the iterate toward a particular point on the Pareto front. Sweeping $\boldsymbol{\lambda}^{(t)}$ within a single run (or across runs) traces out the front.

Pareto MTL [29]. A small set of reference rays $\{\boldsymbol{\lambda}^{(k)}\}$ partitions the simplex into cones; for each ray, training is conducted under a constrained scalarized subproblem with $\boldsymbol{\lambda}^{(t)}$ fixed inside the cone. The descent direction at each step is (17) with $\boldsymbol{\lambda}^{(t)}$ supplied by the active ray.

PaLoRA [10]. Trains a single conditional model on per-step preferences $\boldsymbol{\lambda}^{(t)} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, mixing per-objective LoRA adapters by $\boldsymbol{\lambda}^{(t)}$. The training loss at each step is $\sum_i \lambda_i^{(t)} \ell_i$, and its gradient on $\mathbb{R}^d$ is (17).

PaMaL [9] and FERERO [4]. Sweep $\boldsymbol{\lambda}^{(t)}$ within a single run to trace the Pareto front via preference-conditioned model surgery. The user-specified $\boldsymbol{\lambda}^{(t)}$ enters the update as the linear-scalarization coefficients of (17).

B.4 Common Property: Reconciled Directions Are Euclidean

Across all three families, $\boldsymbol{\lambda}^{(t)}$ is determined by a different mechanism (gradient geometry, loss history, or external preference), but the executed direction is the same object: a linear combination of per-objective Euclidean gradients with $\boldsymbol{\theta}$-detached coefficients, equal to the Euclidean gradient of $\ell_{\boldsymbol{\lambda}^{(t)}}$ on $\mathbb{R}^d$ by (18). This is the hypothesis under which Adam's diagonal RMS metric distorts the per-objective effect (Sec. 3.2) and under which MAdam's preference-conditioned curvature $\mathbf{C}_{\boldsymbol{\lambda}^{(t)}}$ in Eq. (10) is the corresponding Fisher object.

C Sign Flips Under Adam's Geometric Mismatch

Sec. 3.2 introduces the geometric mismatch (Prop. 2, Source 2): Adam's diagonal RMS metric $\mathbf{M}_{\text{Adam}}^{(t)}$ distorts the bilinear form $\mathbf{g}_i^\top \mathbf{d}$, so that under the stationarity approximation $\hat{\mathbf{m}}^{(t)} \approx \mathbf{d}^{(t)}$ the per-objective loss change in Eq. (6) can flip from descent ($\Delta\ell_i^{(t)} < 0$) to ascent on tasks the solver labeled aligned with $\mathbf{d}$. This appendix proves that mechanism: Lemma 1 gives a tight necessary-and-sufficient condition under which a positive-definite diagonal metric flips the sign of $\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d}$ relative to the Euclidean reference $\mathbf{g}_i^\top \mathbf{d}$, expressed as a condition-number trigger $\rho(\mathbf{M}) \geq P_i/N_i$ on the per-task alignment ratio. The remark following the lemma instantiates this trigger to Adam, showing that the per-coordinate anisotropy of $\mathbf{M}_{\text{Adam}}^{(t)}$ in trained networks is large enough to satisfy the trigger even for tasks with substantial Euclidean alignment with $\mathbf{d}$.

Working under the stationarity approximation $\hat{\mathbf{m}}^{(t)} \approx \mathbf{d}^{(t)}$, the realized per-objective loss change in Prop. 2 reduces to $-\eta\, \mathbf{g}_i^{(t)\top}(\mathbf{M}_{\text{Adam}}^{(t)})^{-1}\mathbf{d}^{(t)}$. The next lemma characterizes exactly when a positive-definite diagonal metric flips this bilinear form's sign relative to the Euclidean reference $\mathbf{g}_i^{(t)\top}\mathbf{d}^{(t)}$.

Lemma 1 (Sign-Flip Condition)**.** Let $\mathbf{g}_i, \mathbf{d} \in \mathbb{R}^d$ and let $\mathbf{M} = \text{Diag}(m_1, \ldots, m_d)$ be a positive-definite diagonal metric with condition number $\rho(\mathbf{M}) := m_{\max}/m_{\min}$. Define the positive and negative parts of the coordinate-wise product:

$$P_i := \sum_{k:\, g_{ik}d_k > 0} g_{ik}d_k, \qquad N_i := \Big| \sum_{k:\, g_{ik}d_k < 0} g_{ik}d_k \Big|, \tag{22}$$

so that $\mathbf{g}_i^\top \mathbf{d} = P_i - N_i$. Suppose $\mathbf{g}_i^\top \mathbf{d} > 0$ (task i is aligned with $\mathbf{d}$ in the Euclidean metric) and $N_i > 0$. There exists a diagonal $\mathbf{M}$ with condition number ρ such that $\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d} \leq 0$ if and only if

$$\rho \;\geq\; \frac{P_i}{N_i}. \tag{23}$$

Proof. We have $\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d} = \sum_k g_{ik}d_k/m_k$. Partition the coordinates as $\mathcal{I}^+ = \{k : g_{ik}d_k > 0\}$ and $\mathcal{I}^- = \{k : g_{ik}d_k < 0\}$.

(⇐) Suppose $\rho \geq P_i/N_i$. Set $m_k = m_{\max}$ for $k \in \mathcal{I}^+$ and $m_k = m_{\min}$ for $k \in \mathcal{I}^-$. Then

$$\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d} = \frac{P_i}{m_{\max}} - \frac{N_i}{m_{\min}} = \frac{1}{m_{\min}}\left(\frac{P_i}{\rho} - N_i\right) \leq 0,$$

since $P_i/\rho \leq N_i$ by assumption.

(⇒) Suppose $\rho < P_i/N_i$. For any diagonal $\mathbf{M}$ with condition number ρ,

$$\mathbf{g}_i^\top \mathbf{M}^{-1}\mathbf{d} \geq \frac{P_i}{m_{\max}} - \frac{N_i}{m_{\min}} = \frac{P_i - \rho\, N_i}{m_{\max}} > 0,$$

so no sign flip is possible. □

Remark 1. The condition $\rho \geq P_i/N_i$ has a geometric interpretation. The ratio P_i/N_i measures how lopsided the alignment between $\mathbf{g}_i$ and $\mathbf{d}$ is: a large ratio means positive contributions dominate (task i strongly agrees with the reconciled direction), so a proportionally large condition number is needed to flip the sign. Conversely, when task i is only marginally aligned ($P_i \approx N_i$, i.e. $\mathbf{g}_i^\top \mathbf{d} \approx 0$), even a mildly ill-conditioned metric can trigger a flip. For unit-norm vectors with cosine similarity $c = \mathbf{g}_i^\top \mathbf{d}$, the minimum condition number enabling a sign flip satisfies

$$\rho_{\min}(c) = \frac{P_i}{N_i} = \frac{1+c}{1-c}, \qquad P_i = \frac{1+c}{2},\; N_i = \frac{1-c}{2}, \tag{24}$$

achieved when $P_i + N_i = 1$. Notably, when $\mathbf{g}_i = \mathbf{d}$ (single-objective optimization), $c = 1$ and $\rho_{\min}(1) = \infty$: no finite condition number can flip the sign, consistent with $\mathbf{d}^\top \mathbf{M}^{-1}\mathbf{d} > 0$ for any positive-definite $\mathbf{M}$. In MOO, $c < 1$ in general, so $\rho_{\min}(c)$ is finite. In practice, Adam's per-coordinate effective scale $\sqrt{\hat{v}_k}$ is highly anisotropic and routinely spans several orders of magnitude in trained networks [50, 62], so the trigger $\rho \geq P_i/N_i$ is satisfied even for tasks with substantial alignment (c close to 1).

D Toy-Problem Visualizations

This appendix collects the two-dimensional toy experiments that illustrate the weighting mismatch (Source 1) and the geometric mismatch (Source 2) of Sec. 3.2.

D.1 Multi-Phase Weighting Mismatch

This appendix expands the left panel of Fig. 1 into a multi-phase visualization of the weighting mismatch (Source 1, Sec. 3.2). Whereas Fig. 1 (left) shows a single preference switch $\boldsymbol{\lambda}_A \to \boldsymbol{\lambda}_B$, here we drive the same 2D problem through six consecutive phases of alternating preferences $\boldsymbol{\lambda}_A \leftrightarrow \boldsymbol{\lambda}_B$, exposing how Adam's history-averaged second moment compounds across repeated switches while MAdam's preference-conditioned preconditioner re-aligns the update geometry phase by phase.

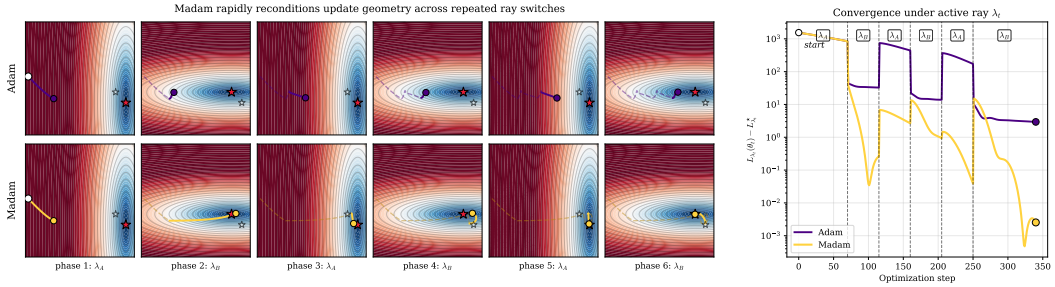


Figure 4: **Multi-phase weighting mismatch: repeated preference switches $\boldsymbol{\lambda}_A \leftrightarrow \boldsymbol{\lambda}_B$.** *Left grid:* decision-space iterates of Adam (top row) and MAdam (bottom row) across six consecutive phases. Each column fixes the active preference $\boldsymbol{\lambda}^{(t)} \in \{\boldsymbol{\lambda}_A, \boldsymbol{\lambda}_B\}$; the background is the corresponding scalarized landscape $\ell_{\boldsymbol{\lambda}^{(t)}}$, the red star marks $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}^{(t)}}$, and the gray star marks the previous-phase optimum. Adam's iterate (purple) lags behind the active optimum after each switch and accumulates bias as phases alternate; MAdam's iterate (yellow) re-converges to the active $\boldsymbol{\theta}^\star_{\boldsymbol{\lambda}^{(t)}}$ within each phase. *Right:* scalarized loss gap $\ell_{\boldsymbol{\lambda}^{(t)}}(\boldsymbol{\theta}^{(t)}) - \ell^\star_{\boldsymbol{\lambda}^{(t)}}$ on the active preference, plotted on a log scale across all six phases. Vertical dashed lines mark phase boundaries. Adam's gap spikes at every switch and plateaus several orders of magnitude above the active optimum, while MAdam drives the gap down in each phase. The contrast confirms Prop. 1: Adam's $\hat{\mathbf{v}}^{(t)}$ tracks the weight-marginalized curvature averaged across past preferences, so the realized metric never matches the active $\boldsymbol{\lambda}^{(t)}$; MAdam's preference-conditioned $\mathbf{C}_{\boldsymbol{\lambda}^{(t)}}$ instantiates the right metric at each step.

D.2 Decision-Space Trajectories Under Rotating Curvature

To make the geometric mismatch of Sec. 3.2 concrete at the optimizer level, we visualize MAdam and Adam trajectories on a two-dimensional toy multi-objective problem with a preference-rotating scalarized metric. The two objectives are matched-scale convex quadratics with opposite principal axes:

$$f_1(\mathbf{x}) = \tfrac{1}{2}(\mathbf{x} - \mathbf{c}_1)^\top \mathbf{H}_1 (\mathbf{x} - \mathbf{c}_1), \quad \mathbf{c}_1 = (1.5,\, 0), \tag{25}$$

$$f_2(\mathbf{x}) = \tfrac{1}{2}(\mathbf{x} - \mathbf{c}_2)^\top \mathbf{H}_2 (\mathbf{x} - \mathbf{c}_2), \quad \mathbf{c}_2 = (-1.5,\, 0). \tag{26}$$

With tilt parameters $\alpha = 2.0$ and $\beta = 0.2$, the Hessians are

$$\mathbf{H}_1 = \begin{bmatrix} 1.1 & 0.9 \\ 0.9 & 1.1 \end{bmatrix}, \qquad \mathbf{H}_2 = \begin{bmatrix} 1.1 & -0.9 \\ -0.9 & 1.1 \end{bmatrix}, \tag{27}$$

so that, written out coordinate-wise,

$$f_1(x_1, x_2) = 0.55\,(x_1 - 1.5)^2 + 0.9\,(x_1 - 1.5)\,x_2 + 0.55\,x_2^2, \tag{28}$$

$$f_2(x_1, x_2) = 0.55\,(x_1 + 1.5)^2 - 0.9\,(x_1 + 1.5)\,x_2 + 0.55\,x_2^2. \tag{29}$$

Both Hessians share the same eigenvalues, so neither task is simply "larger-scale" than the other; the difference is purely directional. The objective f_1 is stiff along the $(1, 1)$ direction, while f_2 is stiff

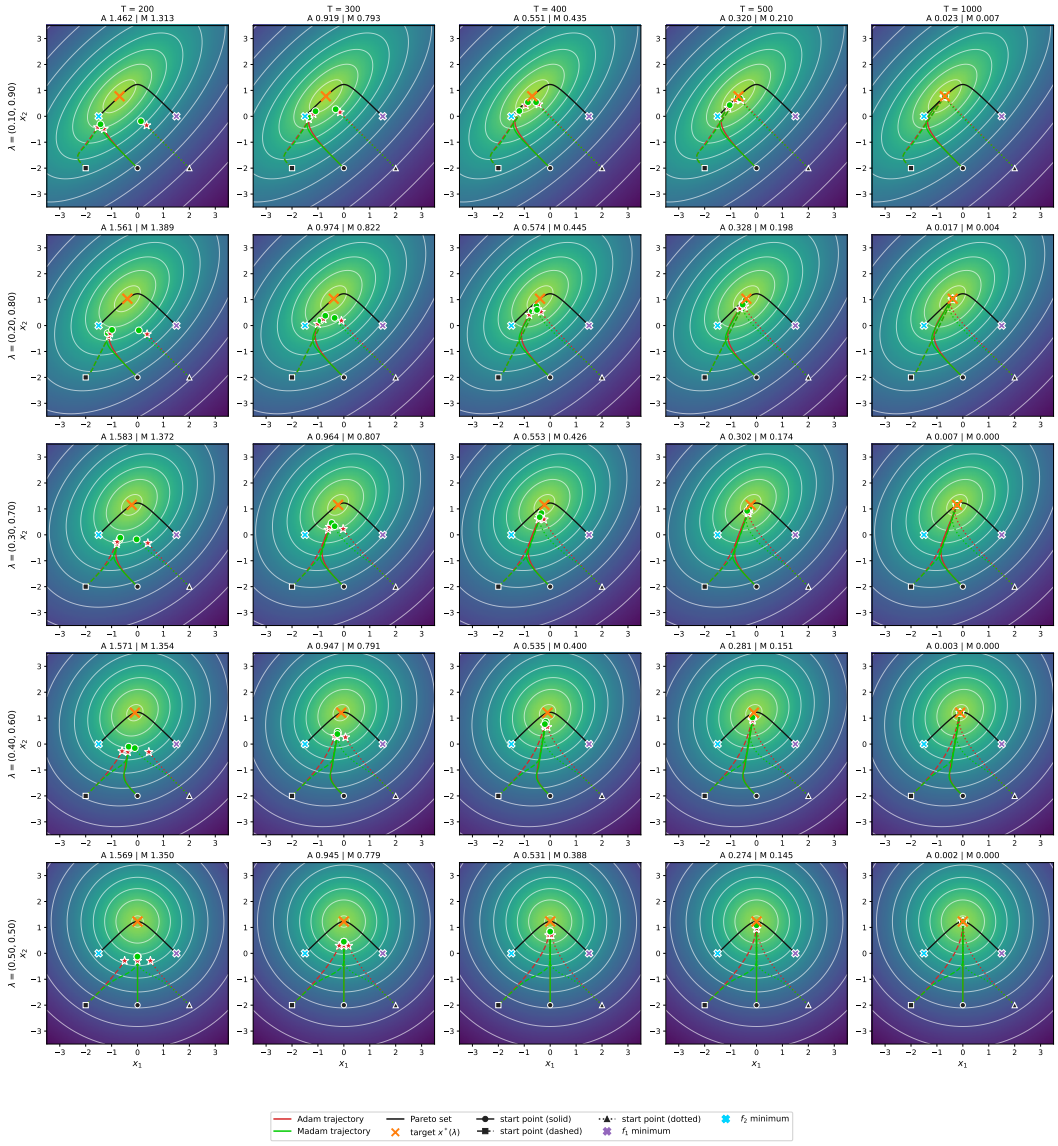


Figure 5: **Decision-space trajectories under changing scalarization geometry.** Each row fixes a preference vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$, and each column shows the optimizer state after a different number of steps. The background in each row is the corresponding scalarized objective $\lambda_1 f_1 + \lambda_2 f_2$ for the pair of matched-scale tilted quadratics defined above. The black curve shows the analytic Pareto set, and the orange cross marks the row-specific scalarized optimum $\mathbf{x}^\star(\boldsymbol{\lambda})$. Red and green trajectories compare Adam and MAdam from three shared initial points. The annotation "A d_{A} | M d_{M}" in each panel (e.g. `A 1.578 | M 1.363`) reports the mean Euclidean distance from the current iterate to the row-specific scalarized optimum $\mathbf{x}^\star(\boldsymbol{\lambda})$, averaged across the three initializations, for Adam (A) and MAdam (M) respectively. Because the two objectives have opposite principal-curvature directions, changing $\boldsymbol{\lambda}$ changes the correct local metric; MAdam more directly tracks this preference-conditioned geometry, while Adam uses coordinate-wise adaptive scaling in the ambient parameter basis.

along $(1, -1)$. Consequently, as the preference vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ varies, the scalarized objective $\lambda_1 f_1 + \lambda_2 f_2$ has its active curvature direction rotated, and the correct local metric rotates with it.

The visualization mirrors the analysis of Sec. 3.2: because the relevant curvature direction rotates with $\boldsymbol{\lambda}$, any coordinate-wise diagonal preconditioner is locked to the ambient basis and cannot follow this rotation, whereas MAdam's preference-conditioned metric adapts to it. The contrast is visible in the trajectories—MAdam bends toward $\mathbf{x}^\star(\boldsymbol{\lambda})$ along the row-specific stiff direction, while Adam takes a path shaped by per-coordinate adaptive scales that do not depend on $\boldsymbol{\lambda}$.

E Experiment Setup: Datasets and Multi-Objectives

This appendix details every benchmark used in Sec. 4: the dataset, the model, the set of objectives $\{\ell_i\}_{i=1}^{C}$ that define the MOO problem, and the metrics reported. Train/validation/test splits and architectures follow the corresponding original references unless noted.

E.1 Multi-Task Learning and Pareto MTL Benchmarks

We adopt the experimental protocol of PaLoRA [10]. For each benchmark we evaluate two regimes: the standard MTL setting, in which a single fixed preference vector $\boldsymbol{\lambda}$ (taken to be uniform unless stated otherwise) is used throughout training, and the Pareto MTL setting, in which a family of preference rays is used to trace out the Pareto front.

`MultiMNIST` **[25, 46].** Each 36×36 input is synthesized by superimposing two MNIST digits, one translated toward the top-left corner and the other toward the bottom-right.

- **Tasks ($C = 2$):** (i) 10-way classification of the top-left digit and (ii) 10-way classification of the bottom-right digit.
- **Losses:** per-objective cross-entropy.
- **Backbone:** a multi-head LeNet comprising a shared convolutional trunk and two classification heads.
- **Metrics:** per-objective top-1 accuracy.
- **Pareto setup:** hypervolume (HV) computed over 11 uniformly spaced preference rays on the 2-simplex, namely $(0.0, 1.0), (0.1, 0.9), \ldots, (1.0, 0.0)$, in the 2-D accuracy space. Reference point: $[0, 0]$.

`SARCOS` **[51].** A canonical inverse-dynamics regression benchmark for the seven degrees of freedom of the `SARCOS` robot arm: given a 21-dimensional state vector consisting of joint positions, velocities, and accelerations, the task is to predict the seven corresponding joint torques.

- **Tasks ($C = 7$):** one regression task per joint torque.
- **Losses:** per-objective mean-squared error.
- **Backbone:** a shared MLP trunk with seven task-specific linear regression heads.
- **Metrics:** per-objective MSE (scaled by a factor of 100 for readability).
- **Pareto setup:** HV computed over 8 preference rays in the 7-D error space: the seven canonical one-hot rays $\mathbf{e}_1, \ldots, \mathbf{e}_7$ together with the uniform ray $(1/7, \ldots, 1/7)$. Reference point: $[1, 5, 1, 1, 10, 10, 1]$.

`UTKFace` **[63].** A collection of aligned and cropped facial images annotated with age, gender, and race.

- **Tasks ($C = 3$):** (i) age regression (continuous, normalized to $[0, 1]$), (ii) binary gender classification, and (iii) 5-way race classification.
- **Losses:** Huber loss for age, binary cross-entropy for gender, and cross-entropy for race.
- **Backbone:** a shared ResNet-18 trunk with three task-specific heads.
- **Metrics:** Huber loss for age (scaled by a factor of 100 for readability) and top-1 accuracy for gender and race.
- **Pareto setup:** the same trained model is queried at each preference vector $\boldsymbol{\lambda} = (\lambda_{\text{age}}, \lambda_{\text{gender}}, \lambda_{\text{race}})$ drawn from a uniform triangulation of the 3-simplex of order 10, yielding 66 preference rays, and HV is computed in the 3-D metric space. Reference point: $[0.5, 0.0, 0.0]$.

`Cityscapes` **[8].** A large-scale urban driving dataset providing dense pixel-level annotations. Following standard practice in the MTL literature, we adopt the downsampled 128×256 resolution variant.

- **Tasks ($C = 2$):** (i) 19-class semantic segmentation and (ii) per-pixel inverse-depth regression.

- **Losses:** pixel-wise cross-entropy for segmentation and an L_1 regression loss for depth.
- **Backbone:** a SegNet-style encoder–decoder architecture with a shared encoder and two task-specific decoders.
- **Metrics:** mean intersection-over-union (mIoU) and pixel accuracy for segmentation; mean absolute and mean relative error for depth.
- **Pareto setup:** the same trained model is queried at each of the 11 uniformly spaced preference rays $(0.0, 1.0), (0.1, 0.9), \ldots, (1.0, 0.0)$ on the 2-simplex, and the best ray under the metric of interest is selected for comparison.

`NYUv2` **[48].** Indoor RGB-D scenes with three dense annotation types.

- **Tasks ($C = 3$):** (i) 13-class semantic segmentation, (ii) per-pixel depth regression, and (iii) surface-normal prediction.
- **Losses:** pixel-wise cross-entropy for segmentation, an L_1 loss for depth, and a cosine distance $1 - \cos\theta$ for surface normals.
- **Backbone:** a SegNet-style encoder–decoder architecture with a shared encoder and three task-specific decoders.
- **Metrics:** mIoU and pixel accuracy for segmentation; mean absolute and mean relative error for depth; mean and median angular error together with the within-threshold accuracies at $\{11.25°, 22.5°, 30°\}$ for normals.
- **Pareto setup:** the same trained model is queried at each preference vector $\boldsymbol{\lambda} = (\lambda_{\text{seg}}, \lambda_{\text{depth}}, \lambda_{\text{normal}})$ drawn from a uniform triangulation of the 3-simplex of order 10, yielding 66 preference rays, and the best ray under the metric of interest is selected for comparison.

E.2 Physics-Informed Neural Networks (PINNs)

We use the PINNacle benchmark [15] with 20 PDE cases spanning Burgers, heat, Poisson, wave, Navier–Stokes, chaotic systems, and high-dimensional Poisson and heat families. For each PDE, the neural network $u_{\boldsymbol{\theta}}$ is trained to satisfy the governing equation in the interior together with the problem-specific boundary, initial, periodic, and observation constraints. Each PDE therefore induces a multi-objective problem whose objectives are the constituent residual and constraint losses, all naturally on different and time-varying scales.

- **Tasks (per PDE):** Each task corresponds to one residual or constraint loss. Depending on the PDE, this includes one or more PDE residual losses ℓ_{PDE} on interior collocation points, boundary-condition losses ℓ_{BC}, initial-condition losses ℓ_{IC}, periodicity losses, and point-observation constraint losses. The number of task losses is PDE-dependent and ranges from 2 to 11 across the active 20 PINNacle cases.
- **Losses:** squared residual or constraint errors evaluated at the corresponding interior, boundary, initial, periodic, or point-observation samples.
- **Backbone:** fully connected MLP $u_{\boldsymbol{\theta}}$ with $\tanh$ activations and Glorot-normal initialization; input and output dimensions are determined by the PDE.
- **Metrics:** relative L_2 error against the available reference solution or reference data, together with final per-component training and test losses.

E.3 Medical Image Analysis

We evaluate MAdam on two medical-imaging benchmarks in which the multi-objective formulation arises from *loss decomposition* rather than from multiple supervised tasks: skin-lesion segmentation on `ISIC2018` and brain-MRI super-resolution on `OASIS3`. In both settings the constituent losses sit on markedly different scales and have qualitatively different gradient distributions, exposing the first-moment lag analyzed in Sec. 3.2 while remaining clean low-dimensional MOO testbeds.

`ISIC2018` **[7].** The ISIC2018 Task 1 lesion-boundary segmentation challenge consists of dermoscopic images paired with binary lesion masks, with a training set of $\sim 2{,}156$ images, a validation set of ~ 538 images and a test set of 1000 images. We follow the standard preprocessing pipeline (resize to a fixed input resolution, intensity normalization).

- **Tasks ($C = 2$):** a single binary lesion-segmentation prediction supervised by two complementary objectives, a pixelwise classification loss and a region-overlap loss.
- **Losses:** pixelwise cross-entropy and soft-Dice loss. Cross-entropy is dominated by easy interior pixels while Dice is dominated by boundary errors, yielding gradients on substantially different scales.
- **Backbones:** UNet [44] and Swin-Transformer [36], trained from scratch.
- **Metrics:** validation Dice coefficient.

`OASIS3` **[24].** A brain-MRI super-resolution benchmark in which a $4\times$ downsampled image is reconstructed to its original resolution. We use a subject-level split of $1{,}129$ volumes for training and 281 for held-out validation. Intensities are clipped to the $[p_1, p_{99}]$ range and min–max normalized to $[0, 1]$; the low-resolution input is generated on the fly by isotropic $4\times$ trilinear downsampling followed by trilinear upsampling back to the original grid.

- **Tasks ($C = 5$):** a single high-resolution reconstruction supervised by one pixelwise loss and four auxiliary losses chosen to capture high-frequency detail and structural fidelity.
- **Losses:** L1 pixelwise reconstruction, Sobel edge, FFT spectral, Laplacian high-pass, and SSIM structural-similarity. The five terms differ in scale by more than an order of magnitude and exhibit distinct gradient distributions.
- **Backbone:** SwinUNETR [16] trained from scratch.
- **Metrics:** LPIPS, DISTS, PSNR and SSIM computed per subject.

F Additional Experimental Results

This appendix collects results that supplement Sec. 4: the NYUv2 MTL table, per-task breakdowns of the Pareto MTL results, per-PDE PINN tables, and the two additional comparisons requested during review (recent MTL solvers and curvature-aware optimizers).

F.1 NYUv2 MTL Results

Tab. 6 reports the full NYUv2 MTL results referenced in the main text.

Table 6: NYUv2 MTL results (mean over three seeds). Surface-normal columns report angle errors (Mean, Med.) and the percentage of pixels within $11.25°/22.5°/30°$. Bold marks the better value within each Base/+MAdam block.

	Seg.		Depth		Surface Normal				
Method	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Mean ↓	Med. ↓	$<11.25°$ ↑	$<22.5°$ ↑	$<30°$ ↑
LS	36.33	63.57	**0.544**	0.222	27.54	21.69	26.59	52.00	64.24
+ Madam	**37.56**	**64.13**	0.550	**0.221**	**27.31**	**21.60**	**26.64**	**52.60**	**64.86**
DWA	37.07	63.55	**0.541**	0.220	**27.63**	**21.90**	26.40	51.59	63.91
+ Madam	**37.21**	**63.89**	**0.541**	**0.218**	**27.63**	22.05	**26.41**	**51.71**	**64.49**
UW	**32.60**	60.87	**0.540**	0.220	**28.22**	**22.43**	**26.05**	**50.60**	**62.69**
+ Madam	32.59	**60.91**	0.546	**0.217**	28.29	22.67	25.55	50.18	62.45
IMTL	36.14	63.51	**0.536**	**0.213**	**25.90**	19.74	29.74	55.98	67.80
+ Madam	**36.26**	**63.68**	0.543	0.214	26.04	**19.70**	**29.84**	**56.03**	**67.81**
CAGrad	**36.99**	**63.90**	**0.543**	**0.216**	25.87	19.72	**29.58**	56.12	67.96
+ Madam	36.88	63.64	0.546	0.224	**25.56**	**19.55**	29.35	**56.32**	**68.39**
PaMaL	**33.94**	**62.55**	0.5592	0.2188	26.60	20.33	29.09	54.61	66.35
+ Madam	33.71	62.34	**0.5540**	**0.2165**	**26.47**	**20.24**	**29.13**	**54.70**	**66.42**
PaLoRA	38.27	64.79	**0.5370**	**0.2150**	25.66	19.34	30.47	56.90	68.56
+ Madam	**38.32**	**64.89**	0.5398	0.2167	**25.60**	**19.28**	**30.57**	**57.07**	**68.75**

F.2 Per-Task Pareto MTL Results

Tabs. 7–9 break the Pareto MTL results of Sec. 4 down by objective and preference ray (mean $\pm$ std over three seeds).

F.2.1 MultiMNIST

Tab. 7 expands the MultiMNIST Pareto fronts of Fig. 2 to a per-objective breakdown for both Pareto-front solvers. For each preference ray $r_0, \dots, r_{10}$ we list the top-left (TL) and bottom-right (BR) digit accuracies separately, together with the 2-D hypervolume HV computed in the two-task accuracy space against the reference point $[0, 0]$. MAdam raises HV from 0.9016 to 0.9177 for PaLoRA and from 0.8869 to 0.9086 for PaMaL, and improves both per-objective accuracies on every ray for PaLoRA. For PaMaL the gain is uniform on the central rays $r_2, \dots, r_8$ but reverses on the extreme rays r_9, r_{10} on the TL task and on r_0 on the BR task: at these endpoints the base solver pushes harder on the dominant task at the expense of the other, whereas MAdam produces a more balanced trade-off, which is consistent with the front-shape change visible in Fig. 2.

F.2.2 SARCOS

Tabs. 8 and 9 expand the SARCOS Pareto MTL results of Tab. 4 to a per-objective breakdown. For each preference ray $r_0, \dots, r_7$ we list the error of every joint task $T_1, \dots, T_7$ (scaled by 100, mean $\pm$ std across three seeds) together with the 7-D hypervolume HV (computed against the reference point $[1, 5, 1, 1, 10, 10, 1]$). PaMaL is markedly less stable than PaLoRA, with occasional seeds blowing up on T_2 and T_5 at extreme rays; the inflated standard deviations in those cells reflect this rather than a systematic regression of MAdam.

Table 7: `MultiMNIST` Pareto MTL results, per-objective breakdown (mean ± std over three seeds). For each base solver and each Base/+MAdam variant we list the top-left (TL) and bottom-right (BR) digit accuracies at every preference ray $r_0, \ldots, r_{10}$, together with HV in the two-task accuracy space.

Method	Task	HV ↑	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
PaMaL	TL	0.8869 ± 0.0060	94.73 ± 0.05	94.54 ± 0.12	94.21 ± 0.10	93.96 ± 0.03	93.75 ± 0.05	93.55 ± 0.11	93.00 ± 0.24	91.89 ± 0.35	89.48 ± 0.27	**85.12 ± 0.82**	**77.44 ± 1.76**
	BR		**76.44 ± 1.81**	83.97 ± 1.14	88.33 ± 0.46	90.68 ± 0.38	91.88 ± 0.49	92.39 ± 0.57	92.57 ± 0.68	92.80 ± 0.78	93.14 ± 0.82	93.50 ± 0.68	93.83 ± 0.50
+ MAdam	TL	**0.9086 ± 0.0050**	**96.01 ± 0.21**	**95.74 ± 0.16**	**95.48 ± 0.24**	**95.30 ± 0.27**	**95.11 ± 0.18**	**94.81 ± 0.12**	**94.16 ± 0.14**	**92.86 ± 0.19**	**89.95 ± 0.22**	83.63 ± 1.58	72.87 ± 4.34
	BR		74.32 ± 2.40	83.87 ± 1.38	**89.15 ± 0.36**	**91.81 ± 0.12**	**93.13 ± 0.36**	**93.70 ± 0.39**	**94.03 ± 0.32**	**94.27 ± 0.39**	**94.51 ± 0.37**	**94.72 ± 0.32**	**94.80 ± 0.24**
PaLoRA	TL	0.9016 ± 0.0045	95.61 ± 0.08	95.63 ± 0.10	95.60 ± 0.11	95.58 ± 0.10	95.48 ± 0.11	95.32 ± 0.12	95.04 ± 0.14	94.71 ± 0.21	94.27 ± 0.24	93.85 ± 0.27	93.20 ± 0.38
	BR		92.02 ± 0.94	92.55 ± 0.76	92.99 ± 0.67	93.26 ± 0.57	93.59 ± 0.45	93.83 ± 0.40	93.99 ± 0.38	94.19 ± 0.35	94.24 ± 0.28	94.22 ± 0.28	94.26 ± 0.27
+ MAdam	TL	**0.9177 ± 0.0024**	**96.45 ± 0.00**	**96.45 ± 0.05**	**96.36 ± 0.11**	**96.27 ± 0.11**	**96.20 ± 0.05**	**96.05 ± 0.08**	**95.82 ± 0.03**	**95.57 ± 0.03**	**95.11 ± 0.14**	**94.61 ± 0.18**	**93.93 ± 0.40**
	BR		**92.71 ± 0.46**	**93.30 ± 0.41**	**93.76 ± 0.35**	**94.06 ± 0.34**	**94.36 ± 0.35**	**94.55 ± 0.34**	**94.72 ± 0.26**	**94.90 ± 0.27**	**94.98 ± 0.27**	**95.06 ± 0.21**	**95.12 ± 0.17**

Table 8: `SARCOS` Pareto MTL results for **PaMaL**, per-objective breakdown (errors ×100, mean ± std over three seeds). HV is computed in the 7-D error space. Bold marks the better value within each Base/+MAdam block (lower is better).

Method	Task	HV ↑	r_0 ↓	r_1 ↓	r_2 ↓	r_3 ↓	r_4 ↓	r_5 ↓	r_6 ↓	r_7 ↓
PaMaL	T1	378.29 ± 9.27	**2.55 ± 0.13**	8.15 ± 2.21	**6.88 ± 2.15**	6.24 ± 0.93	7.63 ± 1.07	**7.08 ± 1.34**	6.69 ± 1.67	5.72 ± 0.49
	T2		99.65 ± 31.18	35.68 ± 2.17	**60.41 ± 4.89**	80.37 ± 33.37	248.11 ± 194.08	63.79 ± 8.22	73.86 ± 5.44	46.55 ± 1.69
	T3		4.50 ± 1.53	3.03 ± 0.36	**1.19 ± 0.07**	2.67 ± 0.39	5.49 ± 1.45	2.51 ± 0.15	3.13 ± 0.70	1.79 ± 0.04
	T4		3.04 ± 0.53	2.69 ± 0.59	**1.97 ± 0.41**	0.57 ± 0.26	5.11 ± 3.03	1.89 ± 0.35	2.44 ± 1.02	1.35 ± 0.06
	T5		470.92 ± 468.44	197.50 ± 46.96	**142.53 ± 13.23**	**139.74 ± 6.40**	117.39 ± 26.31	181.94 ± 37.38	230.66 ± 75.37	114.95 ± 2.87
	T6		**32.20 ± 2.91**	39.14 ± 2.98	**29.49 ± 1.90**	27.79 ± 1.94	44.53 ± 5.93	22.75 ± 7.90	28.77 ± 1.19	25.75 ± 0.72
	T7		2.13 ± 0.22	**2.35 ± 0.53**	**1.90 ± 0.55**	**1.63 ± 0.09**	3.08 ± 1.48	3.01 ± 1.31	41.40 ± 57.86	1.24 ± 0.04
+ MAdam	T1	**394.55 ± 4.20**	3.20 ± 1.51	**6.60 ± 0.70**	9.03 ± 5.98	**5.23 ± 0.58**	**6.95 ± 1.43**	7.53 ± 0.42	**4.96 ± 0.39**	**4.95 ± 0.06**
	T2		**65.91 ± 18.98**	**29.59 ± 0.23**	236.04 ± 251.90	**68.36 ± 20.81**	**86.71 ± 5.40**	**55.37 ± 5.25**	**59.14 ± 4.04**	**40.47 ± 3.63**
	T3		**2.59 ± 0.42**	**2.91 ± 0.15**	2.27 ± 1.63	**2.29 ± 0.26**	**3.79 ± 1.56**	**2.29 ± 0.14**	**2.25 ± 0.17**	**1.63 ± 0.10**
	T4		**2.04 ± 0.80**	**2.21 ± 0.33**	4.84 ± 3.38	**0.29 ± 0.02**	**2.72 ± 0.67**	**1.73 ± 0.14**	**1.82 ± 0.99**	**1.09 ± 0.05**
	T5		**115.65 ± 5.26**	**152.62 ± 9.57**	312.65 ± 240.60	144.33 ± 10.84	**90.34 ± 1.34**	**128.65 ± 9.42**	**131.23 ± 15.43**	**104.75 ± 3.07**
	T6		32.97 ± 3.82	**34.65 ± 2.61**	33.92 ± 8.04	**24.98 ± 1.81**	**37.56 ± 2.52**	**16.06 ± 2.13**	**28.15 ± 1.50**	**21.17 ± 0.29**
	T7		**2.05 ± 0.33**	2.78 ± 0.54	4.29 ± 2.91	1.70 ± 0.41	**2.38 ± 0.27**	**2.48 ± 0.97**	**1.45 ± 0.78**	**1.00 ± 0.05**

Table 9: `SARCOS` Pareto MTL results for **PaLoRA**, per-objective breakdown (errors ×100, mean ± std over three seeds). HV is computed in the 7-D error space. Bold marks the better value within each Base/+MAdam block (lower is better; ties to two decimals are left unbolded).

Method	Task	HV ↑	r_0 ↓	r_1 ↓	r_2 ↓	r_3 ↓	r_4 ↓	r_5 ↓	r_6 ↓	r_7 ↓
PaLoRA	T1	431.21 ± 8.47	2.00 ± 0.14	3.15 ± 1.19	2.90 ± 1.18	2.65 ± 0.64	2.53 ± 0.34	2.27 ± 0.26	2.52 ± 0.66	2.16 ± 0.21
	T2		30.73 ± 9.03	22.06 ± 7.79	34.19 ± 19.67	23.56 ± 2.36	29.87 ± 5.78	32.99 ± 10.61	30.93 ± 9.54	21.17 ± 3.20
	T3		1.09 ± 0.32	**1.32 ± 0.46**	**0.92 ± 0.22**	**0.95 ± 0.12**	**1.06 ± 0.07**	**0.93 ± 0.14**	**1.13 ± 0.42**	**0.94 ± 0.18**
	T4		0.43 ± 0.18	**0.36 ± 0.03**	0.34 ± 0.07	0.30 ± 0.04	0.34 ± 0.04	0.39 ± 0.11	0.37 ± 0.10	0.31 ± 0.05
	T5		67.43 ± 7.00	69.84 ± 4.36	73.76 ± 17.93	69.35 ± 6.04	62.33 ± 5.38	84.22 ± 32.89	82.23 ± 21.24	60.15 ± 2.55
	T6		13.47 ± 2.60	14.03 ± 1.88	**15.55 ± 3.16**	**11.49 ± 0.95**	**11.75 ± 0.56**	**10.37 ± 0.58**	**12.09 ± 1.44**	**11.14 ± 1.16**
	T7		0.57 ± 0.11	1.31 ± 1.28	0.52 ± 0.14	0.41 ± 0.04	0.42 ± 0.06	0.83 ± 0.59	0.40 ± 0.07	0.44 ± 0.10
+ MAdam	T1	**442.17 ± 3.41**	**1.59 ± 0.19**	**1.93 ± 0.05**	**1.91 ± 0.08**	**1.83 ± 0.01**	**1.88 ± 0.13**	**1.85 ± 0.05**	**1.99 ± 0.31**	**1.67 ± 0.04**
	T2		**22.54 ± 3.75**	**15.60 ± 0.82**	**20.00 ± 4.16**	**17.60 ± 0.29**	**21.25 ± 2.89**	**19.87 ± 0.84**	**18.77 ± 0.68**	**16.47 ± 1.12**
	T3		**0.95 ± 0.30**	1.36 ± 0.51	1.34 ± 0.97	1.22 ± 0.66	1.41 ± 0.74	1.39 ± 0.76	1.28 ± 0.70	1.17 ± 0.67
	T4		**0.32 ± 0.06**	0.38 ± 0.11	**0.30 ± 0.01**	**0.24 ± 0.03**	**0.31 ± 0.05**	**0.29 ± 0.04**	**0.28 ± 0.01**	**0.24 ± 0.00**
	T5		**64.73 ± 15.49**	**64.23 ± 7.92**	**54.06 ± 5.05**	**53.67 ± 2.84**	**50.14 ± 0.94**	**71.29 ± 7.43**	**67.00 ± 18.93**	**52.35 ± 4.06**
	T6		**13.30 ± 2.47**	**13.66 ± 2.82**	16.70 ± 9.04	12.20 ± 2.13	22.96 ± 14.33	11.79 ± 3.73	14.49 ± 5.21	13.91 ± 4.56
	T7		**0.38 ± 0.03**	**0.58 ± 0.26**	**0.41 ± 0.02**	0.41 ± 0.09	**0.38 ± 0.02**	**0.51 ± 0.16**	**0.32 ± 0.03**	**0.37 ± 0.05**

F.3 PINN Results

Tabs. 10–12 report the per-PDE mean L2RE on PINNacle for the three baseline families summarized in Sec. 4, each paired with MAdam.

Table 10: Mean L2RE on the **Vanilla** PINN family (PINN). Bold marks the better Base/+MAdam value (lower is better).

Method	Burgers		Poisson			Heat				NS			Wave			Chaotic		High dim	
	1d-C	2d-C	2d-C	3d-CG	2d-MS	2d-VC	2d-MS	2d-CG	2d-LT	2d-C	2d-CG	2d-LT	1d-C	2d-CG	2d-MS	GS	KS	PNd	HNd
PINN [43]	1.45E-2	**3.24E-1**	6.94E-1	5.60E-1	6.30E-1	1.01E+0	6.21E-2	3.64E-2	9.99E-1	4.70E-2	1.19E-1	9.96E-1	5.88E-1	1.84E+0	1.34E+0	3.19E-1	1.01E+0	3.04E-3	**3.61E-1**
+ MAdam	**1.29E-2**	5.16E-1	**6.62E-1**	**1.94E-1**	**6.25E-1**	**9.40E-1**	**2.83E-2**	**1.82E-2**	**9.98E-1**	**4.15E-2**	**9.74E-2**	**9.94E-1**	**4.86E-1**	**1.00E+0**	**8.05E-1**	**9.32E-2**	**9.71E-1**	**7.86E-4**	5.46E-1

Table 11: Mean L2RE on the **Loss Reweighting/Sampling** PINN family (LRA, NTK, RAR). Bold marks the better Base/+MAdam value (lower is better); NaN denotes non-convergence.

Method	Burgers		Poisson			Heat				NS			Wave			Chaotic		High dim	
	1d-C	2d-C	2d-C	3d-CG	2d-MS	2d-VC	2d-MS	2d-CG	2d-LT	2d-C	2d-CG	2d-LT	1d-C	2d-CG	2d-MS	GS	KS	PNd	HNd
LRA [53]	2.61E-2	**2.60E-1**	1.17E-1	1.02E-1	7.94E-1	2.12E-1	8.79E-2	1.25E-1	9.99E-1	NaN	3.32E-1	1.00E+0	3.61E-1	1.48E+0	1.02E+0	**9.37E-2**	9.57E-1	4.58E-4	3.94E-1
+ MAdam	**2.04E-2**	4.64E-1	**2.62E-2**	**8.47E-2**	**7.11E-1**	**2.01E-1**	**3.93E-2**	**1.08E-1**	9.99E-1	NaN	**3.01E-1**	1.00E+0	**1.25E-1**	**1.01E+0**	**9.96E-1**	1.24E-1	**9.52E-1**	4.58E-4	**2.85E-1**
NTK [54]	1.84E-2	**2.75E-1**	1.23E-2	9.47E-1	7.48E-1	2.14E-1	4.40E-2	1.16E-1	1.00E+0	1.98E-1	2.93E-1	9.99E-1	9.79E-2	2.16E+0	1.04E+0	2.16E-1	9.64E-1	**4.64E-3**	3.97E-1
+ MAdam	**1.48E-2**	4.64E-1	**5.60E-3**	**9.41E-1**	**7.18E-1**	**2.00E-1**	**2.41E-2**	**8.71E-2**	**9.98E-1**	**1.76E-1**	**2.80E-1**	9.99E-1	**8.51E-2**	**9.90E-1**	**7.91E-1**	**1.85E-1**	**9.47E-1**	4.74E-3	**3.92E-1**
RAR [56]	3.32E-2	**3.45E-1**	6.99E-1	5.76E-1	**6.44E-1**	9.66E-1	7.49E-2	2.72E-2	9.99E-1	4.69E-1	3.34E-1	1.00E+0	5.39E-1	1.15E+0	1.35E+0	**9.46E-2**	1.01E+0	3.59E-3	**3.57E-1**
+ MAdam	**2.43E-2**	5.21E-1	**6.90E-1**	**5.12E-1**	6.51E-1	**9.63E-1**	**3.18E-2**	**1.84E-2**	9.99E-1	**4.08E-2**	**1.02E-1**	**9.94E-1**	**4.96E-1**	**1.04E+0**	**9.09E-1**	1.93E-1	1.01E+0	**3.21E-3**	3.81E-1

Table 12: Mean L2RE on the **Architecture** PINN family (LAAF, GAAF). Bold marks the better Base/+MAdam value (lower is better).

Method	Burgers		Poisson			Heat				NS			Wave			Chaotic		High dim	
	1d-C	2d-C	2d-C	3d-CG	2d-MS	2d-VC	2d-MS	2d-CG	2d-LT	2d-C	2d-CG	2d-LT	1d-C	2d-CG	2d-MS	GS	KS	PNd	HNd
LAAF [19]	1.43E-2	**2.77E-1**	7.68E-1	5.79E-1	5.93E-1	**6.42E-1**	**7.40E-2**	**2.39E-2**	9.99E-1	3.60E-2	8.24E-2	9.98E-1	4.54E-1	8.19E-1	1.06E+0	**9.47E-2**	1.01E+0	4.14E-3	5.22E-1
+ MAdam	**1.29E-2**	4.94E-1	**6.35E-1**	**2.04E-1**	**4.69E-1**	8.66E-1	1.01E-1	2.45E-2	**9.98E-1**	**2.13E-2**	**6.63E-2**	**9.97E-1**	**4.20E-1**	**7.62E-1**	**1.03E+0**	9.48E-2	**9.98E-1**	**4.10E-3**	**4.96E-1**
GAAF	5.20E-2	**2.95E-1**	**6.04E-1**	5.02E-1	**9.31E-1**	**8.49E-1**	9.85E-1	4.61E-1	**9.99E-1**	3.79E-2	1.74E-1	9.99E-1	6.77E-1	7.94E-1	1.06E+0	9.46E-2	**1.00E+0**	7.75E-3	5.21E-1
+ MAdam	**2.01E-2**	5.08E-1	6.06E-1	**2.30E-1**	9.79E-1	1.06E+0	**1.26E-1**	**4.60E-1**	1.00E+0	**3.62E-2**	**1.67E-1**	9.99E-1	**6.59E-1**	**7.24E-1**	**1.03E+0**	**9.45E-2**	1.01E+0	**7.70E-3**	**4.85E-1**

F.4 Curvature-Aware Optimizers

To separate the effect of the preference-conditioned metric from that of curvature information in general, we compare MAdam against curvature-aware optimizers on SARCOS under LS, using the setup of Tab. 2 and three seeds. FAdam [18] preconditions by a diagonal empirical Fisher estimated from the update direction, Shampoo [14] maintains full preconditioners along each tensor axis, and AdaHessian [59] estimates the Hessian diagonal by Hutchinson sampling. All optimizers use the Adam hyperparameters of the baseline without optimizer-specific retuning; the Adam and MAdam rows coincide with the LS block of Tab. 2. Tab. 13 reports the results. Adam and FAdam perform similarly: FAdam estimates its Fisher from the scalarized direction $\mathbf{d}^{(t)}$ and therefore inherits the weight marginalization of Prop. 1. Shampoo and AdaHessian degrade substantially under the shared hyperparameters, indicating that they require optimizer-specific calibration to transfer to this setting. MAdam reuses the Adam configuration and reduces the average error from 10.99 to 10.00, which attributes the gain to the preference-conditioned cross-objective curvature rather than to curvature information alone.

Table 13: SARCOS MTL results under LS for curvature-aware optimizers (mean over three seeds). Errors are scaled by 100. Bold marks the best value in each column.

Optimizer	Avg. err. ↓	t_1 ↓	t_2 ↓	t_3 ↓	t_4 ↓	t_5 ↓	t_6 ↓	t_7 ↓
Adam	10.99	1.63	14.87	0.65	**0.19**	50.32	8.96	**0.30**
FAdam [18]	11.40	1.65	15.16	0.70	0.20	52.38	9.44	0.31
Shampoo [14]	41.33	3.61	73.91	1.47	0.28	183.45	26.20	0.39
AdaHessian [59]	99.01	10.27	238.84	5.49	4.00	366.31	64.31	3.84
MAdam	**10.00**	**1.46**	**13.55**	**0.61**	**0.19**	**45.43**	**8.47**	0.31

G Computational Cost

Per-step overhead. We profile MAdam on SARCOS with an NVIDIA A6000 GPU, using five paired profiles per method of the same training run with and without MAdam. Tab. 14 reports steady-state wall-clock time per step and peak allocated memory for LS (loss balancing) and CAGrad (gradient balancing). The added wall-clock time comes from the two sampled per-objective backward passes of Sec. 3.5 and is similar in absolute terms for both solvers (about 7 ms); the relative increase is smaller for CAGrad because its base step already computes per-objective gradients and is therefore more expensive. The added memory holds the $C(C+1)/2$ cross-objective Fisher EMAs of Eq. (15), which scale as $O(C^2)$ in the number of objectives and linearly in the shared-parameter dimension. Only the number of sampled per-objective gradient evaluations per step is independent of C; total runtime and memory are not.

Table 14: Per-step cost on SARCOS ($C = 7$; NVIDIA A6000; mean over five paired profiles). Wall-clock is per optimizer step; peak memory is peak allocated GPU memory.

	Wall-clock (ms/step)			Peak memory (MiB)	
Solver	Base	+MAdam	Δ	Base	+MAdam
LS	36.09	43.00	$+19.3\%$	20.98	39.91
CAGrad	56.59	63.33	$+12.1\%$	31.60	46.78

Matched-quality cost. Per-step overhead does not capture the total computation required to reach a given solution quality. Tab. 15 compares CAGrad under Adam and under MAdam on SARCOS at a matched validation error of 0.479 ± 0.052 (mean $\pm$ std over three seeds), reporting the number of steps and the cumulative FLOPs needed to reach it. MAdam reaches the target in 33% fewer steps and with 20% fewer cumulative FLOPs, so its faster convergence offsets the per-step overhead in this comparison.

Table 15: Steps and cumulative compute to reach a matched validation error of 0.479 ± 0.052 for CAGrad on SARCOS (mean $\pm$ std over three seeds).

Method	Steps to target ↓	Cumulative TFLOPs ↓
CAGrad + Adam	$5{,}965 \pm 1{,}041$	14.10 ± 2.46
CAGrad + MAdam	$\mathbf{4{,}009 \pm 1{,}084}$	$\mathbf{11.31 \pm 3.04}$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main claims and contributions of the paper.

   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 the Conclusion section.

   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 paper provides assumptions and complete proofs in appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide annoymized code link.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use public datasets and provide code link.

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 provide details in the main paper and appendix, with the baseline we following.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report standard deviation across multiple benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the computer hardware details for experiment.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discussed the social impact of our work topic Multi-objective optimization in the Introduction and Conclusion sections.

    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: No such data or models are released in the paper.

    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 correctly cited the original paper and all baselines.

    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: [Yes]

    Justification: We have provided readme for the code.

    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: Not apply.

    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: Not apply.

    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 use LLM for improve writing only.

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.