Information Transfer Across Clinical Tasks via Adaptive Parameter Optimisation

Anshul Thakur[1] **Elena Gal**[1] **Soheila Molaei**[1] **Xiao Gu**[1]
Patrick Schwab[2] **Danielle Belgrave**[2] **Kim Branson**[2] **David A. Clifton**[1,3]
[1]Institute of Biomedical Engineering, University of Oxford, UK
[2]GlaxoSmithKline,London, UK
[3]Oxford-Suzhou Institute of Advanced Research (OSCAR), Suzhou, China

Abstract

This paper presents Adaptive Parameter Optimisation (APO), a novel framework for optimising shared models across multiple clinical tasks, addressing the challenges of balancing strict parameter sharingoften leading to task conflictsand soft parameter sharing, which may limit effective cross-task information exchange. The proposed APO framework leverages insights from the lazy behaviour observed in over-parameterised neural networks, where only a small subset of parameters undergo any substantial updates during training. APO dynamically identifies and updates task-specific parameters while treating parameters associated with other tasks as protected, limiting their modification to prevent interference. The remaining unassigned parameters remain unchanged, embodying the lazy training phenomenon. This dynamic management of task-specific, protected, and unclaimed parameters across tasks enables effective information sharing, preserves task-specific adaptability, and mitigates gradient conflicts without enforcing a uniform representation. Experimental results across diverse healthcare datasets demonstrate that APO surpasses traditional information-sharing approaches, such as multi-task learning and model-agnostic meta-learning, in improving task performance.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

1 INTRODUCTION

Deep learning holds transformative potential in healthcare, particularly in diagnostics, personalised medicine, and patient management (Liu et al., 2019; Wilkinson et al., 2020; Rajkomar et al., 2018). However, a major obstacle to realizing this potential is the fragmented nature of healthcare data, largely due to privacy regulations like the Data Protection Act (DPA), which result in siloed Electronic Health Records (EHRs) across institutions (Thakur et al., 2021). This fragmentation, combined with significant data heterogeneity, limited labels, and imbalanced datasets, forces clinical models to rely on site- or population-specific data, which limits their generalisability. Notably, many clinical taskssuch as predicting sepsis, patient deterioration, or mortalityshare overlapping clinical features or are semantically related. This interconnectedness offers an opportunity to enhance generalisation through information-sharing mechanisms, particularly in data-scarce settings where limited data and heterogeneity can hinder model performance (Caruana, 1997).

Traditional information-sharing mechanisms like Multi-Task Learning (MTL) and Model-Agnostic Meta-Learning (MAML) aim to optimise shared models across tasks by aggregating gradients from all tasks (Allenspach et al., 2024; Finn et al., 2017). These approaches attempt to capture common features in a shared representation space. While effective for closely related tasks, they often struggle when tasks exhibit weak or conflicting associations, resulting in suboptimal performance due to negative transfer (Yu et al., 2020). In such cases, gradient interference among tasks hampers the learning of a shared representation, making convergence difficult and resulting in performance degradation compared to task-specific models.

In clinical settings, while tasks often share similar in-

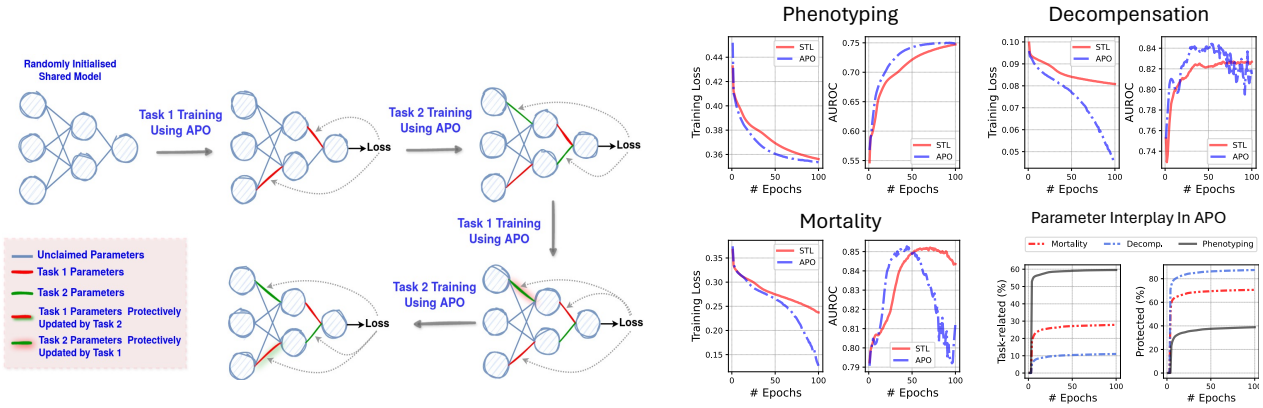


Figure 1: ***Left*** Collaboration in Adaptive Parameter Optimisation (APO): Sensitive parameters are selected, with task-specific or unclaimed ones updated normally. Parameters linked to other tasks undergo protected updates using proximal regularisation to induce minimal changes. ***Right*** Training dynamics of MIMIC-III tasks using APO and single-task learning (STL), including validation scores, as well as the evolution of task-specific and protected parameters in the shared model.

put features, they vary in frequency and the specific outcomes they target. For instance, mortality prediction may rely on ICU data from the first 48 hours, whereas respiratory deterioration prediction requires continuous monitoring (Johnson et al., 2016; Harutyunyan et al., 2019). This variation in task timing creates challenges for hard parameter sharing, as frequent tasks (e.g., respiratory decompensation) may be overshadowed by less frequent tasks (e.g., mortality prediction), leading to suboptimal performance. Although methods like GradNorm (Chen et al., 2018) and gradient projection (Yu et al., 2020) have been developed to mitigate these conflicts, they can still fall short when task dynamics are highly unbalanced. As a more flexible alternative, soft parameter sharing reduces the rigidity of hard sharing by training tasks independently while constraining models to remain close to each other (Ruder, 2017). However, this approach can limit effective information transfer between tasks, thus reducing the potential benefits of joint optimisation. Consequently, there remains a need for a mechanism that balances task-specific flexibility, knowledge sharing, and the mitigation of gradient conflicts, particularly in settings where tasks share representations but differ in frequency and complexity.

To address this need, this paper introduces Adaptive Parameter Optimisation (APO), a novel framework that overcomes the limitations of both hard and soft parameter sharing. Instead of enforcing a uniform representation across tasks, APO dynamically adapts to the evolving requirements of each task. Inspired by the lazy behaviour of over-parameterised networks, where only a subset of parameters deviate from their initial state during training (Chizat et al., 2019; Thakur et al., 2022), APO partitions shared layer parameters into three categories: task-specific (sensitive to the tasks training loss), protective (associated with other tasks), and unclaimed (not yet influenced by any task). During each iteration, tasks dynamically identify sensitive parameters based on their training loss, and only these sensitive parameters are eligible for updates. From this set of sensitive parameters, task-specific and unclaimed parameters are updated as usual, while protective parameters are adjusted with a proximal regularisation term to minimise interference with other tasks and maintain cross-task knowledge transfer. Figure 1 provides a simplified illustration of the entire process.

This fine-grained control over parameter allocation promotes effective knowledge sharing while reducing the conflicts typically seen in hard parameter sharing. Unlike soft parameter sharing, which keeps models closely aligned through regularisation, APO allows tasks to evolve independently while still benefiting from shared information. By dynamically conditioning each tasks learning on the parameter updates of others, APO enables task-specific adaptations without rigid constraints, effectively balancing task independence with cross-task knowledge transfer. Moreover, APO introduces a natural form of data-dependent regularisation by updating only sensitive parameters at each step. The number of these sensitive parameters gradually increases as needed, modulating the model's effective degrees of freedom (Figure 1, *right panel*). As a result, APO facilitates information transfer while introducing implicit regularisationtwo key phenomena associated with joint optimisationresulting in improved sample efficiency and generalised models.

This paradigm is particularly advantageous in clinical settings, where tasks often share common characteris-

tics but require specialised treatment of their unique features. For instance, clinical tasks like mortality prediction, respiratory decompensation, and patient deterioration prediction may overlap in some features but need task-specific focus to capture their subtleties. APO excels in these scenarios by allowing tasks to share relevant information while preserving the specificity crucial for each case. As a result, APO adapts to the complexities and diverse relationships of clinical tasks, enabling the development of scalable, generalised models that are crucial for deployment in real-world clinical settings.

2 EARLIER STUDIES

Multi-task learning: Most MTL studies address gradient conflicts and negative transfer to optimise shared models effectively. These methods often focus on architectural enhancements and optimisation strategies. For example, Cross-Stitch (Misra et al., 2016) facilitates learning by dynamically combining task-specific representations, while the multi-gate mixture-of-experts (MMoE) (Ma et al., 2018) approach shares experts across tasks, using a gating mechanism to select task-specific experts. Building on MMoE, Chen et al. (Chen et al., 2023) replaced the gating mechanism with a more flexible expert selection process based on both input and task characteristics. Deviating from architectural improvements, optimisation strategies in MTL often combat negative transfer by adapting loss weights or directly adjusting gradients. For instance, Kendall et al. (Kendall et al., 2018) introduced task-specific loss weighting using homoscedastic uncertainty, which inspired methods like GradNorm (Chen et al., 2018), adjusting gradients based on task learning rates. Other approaches, such as gradient projection (Yu et al., 2020), resolve conflicts by modifying gradients. Navon et al. framed MTL as a bargaining game, using Nash equilibrium to balance gradients, ensuring fair optimisation across tasks without any task dominating (Navon et al., 2022). Deviating from these strategies, Mirzadeh et al. (Mirzadeh et al., 2021) perform MTL by finding connected solutions for tasks through low-loss paths or linear mode connections, which minimises gradient conflicts and improves the optimisation process across tasks.

Meta-Learning: Gradient-based meta-learning approaches, such as MAML and its variants (iMAML (Rajeswaran et al., 2019), REPTILE (Nichol and Schulman, 2018), and ANIL (Raghu et al., 2019)), aim to learn a meta-initialisation or shared global model across tasks, enabling fast adaptation to unseen tasks with few gradient updates. This shared global model is analogous to MTL model, making meta-learning applicable to multi-task frameworks (Thakur et al., 2021). However, as joint optimisation in both MTL and meta-learning is identical (Wang et al., 2021), both face challenges like negative transfer and gradient conflicts in presence of divergent tasks. To overcome this issue, multi-modal meta-learning methods (Vuorio et al., 2019; Abdollahzadeh et al., 2021) have been designed to handle such scenarios. These methods work under the assumption that meta-training tasks are from different modes and exploit a mode-specific parameter modulation mechanism to align the global model parameters with the task mode.

Comparison with the proposed framework: In comparison to MTL and standard meta-learning approaches, the proposed APO framework avoids enforcing a shared representation across tasks, allowing each task to adapt the model as needed, thereby alleviating gradient conflicts. Unlike multi-modal meta-learning, APO does not rely on complex gradient aggregation or specialised architectures, instead leveraging standard SGD to handle task adaptability and flexibility effectively.

3 MOTIVATION

In deep learning, the phenomenon of *lazy training* refers to the observation in over-parameterised neural networks where only a small subset of parameters exhibit any deviation from their initial random state to achieve effective convergence (Li and Liang, 2018; Allen-Zhu et al., 2019). Chizat *et al.* later demonstrated that this behaviour is not unique to over-parameterised networks; it can also be induced in any network through specific choices of hyperparameters, such as normalisation, initialisation, and the number of training iterations (Chizat et al., 2019). According to their study, lazy behaviour is characterised by a scenario where a small relative change in the model parameters, $\Delta\theta$, leads to a large relative change in the objective or loss, $\Delta\ell$, when a model with parameters θ is updated to $\hat{\theta}$ using a loss function ℓ:

$$\Delta\ell = \frac{\ell(\theta) - \ell(\hat{\theta})}{\ell(\theta)} \gg \Delta\theta = \frac{\|\theta - \hat{\theta}\|}{\|\theta\|}. \tag{1}$$

This smaller relative change in the overall model can result either from minor deviations across all parameters or from significant deviations in only a few parameters. Most over-parameterised networks naturally exhibit the latter behaviour, which is also the focus of this work.

The motivation behind the proposed APO framework is to explicitly harness this characteristic of lazy training by dynamically selecting and updating only a small subset of parameters that are most sensitive to the

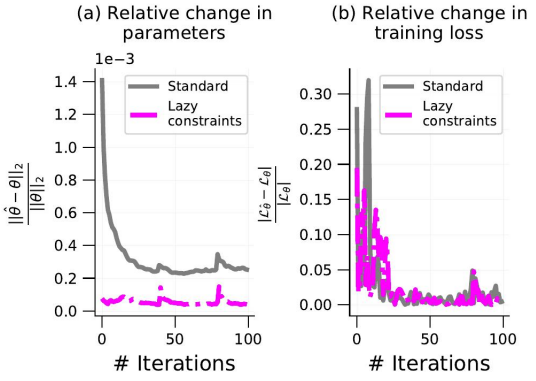


Figure 2: Relative change in (a) parameters and (b) training loss observed during the first 100 iterations of training ResNet-101 on CIFAR-10 using standard and lazy constraints based training.

training loss. Sensitivity, in this context, refers to the magnitude of the gradient of a parameter. By limiting updates to this subset of highly sensitive parameters, APO effectively induces a lazy training regime that leads to a smaller relative change in the overall model while still achieving a meaningful impact on the training objective.

This selective approach offers two key advantages: first, it ensures that only the most influential parameters are updated, thereby amplifying the effect on the objective function; second, it minimises unnecessary changes to other parameters, reducing the risk of gradient conflicts and preserving information relevant to other tasks. Figure 2 illustrates this effect by comparing the relative changes in loss and parameters when training a ResNet-101 model under both normal and induced lazy scenarios. In the lazy scenario, only the top 1% of the most sensitive parameters are updated in each iteration, leading to efficient convergence with minimal parameter adjustment. This strategy not only drives efficient convergence but also lays the groundwork for robust joint optimisation by balancing task-specific adaptation with the stability needed to mitigate interference across tasks.

4 PROPOSED METHOD

4.1 Notations and symbols

Let $f_{\boldsymbol{\theta}}(\cdot)$ be a shared neural network where $\boldsymbol{\theta} \in \mathbb{R}^N$ is a trainable tensor and N is the total trainable parameters. $\mathcal{L}^t_{\boldsymbol{\theta}}(\cdot)$ is an optimisation objective for task t. $\mathbf{G} \in \mathbb{R}^N$ is an "accumulator variable" used to accumulate the historic gradients throughout training for every parameter in $\boldsymbol{\theta}$. $\mathbf{H}^t \in \mathbb{R}^N$ is a mask highlighting the parameters associated with task t, and a global mask $\mathbf{P} \in \mathbb{R}^N$ to track of all parameters updated during the training. $\mathbf{G}$, $\mathbf{H}^t$ and $\mathbf{P}$ are initialised with zero tensor. $\odot$, $\wedge$, $\vee$ and $\neg$ represent element-wise multiplication, logical AND, logical OR and logical NOT operations, respectively.

Algorithm 1 Proposed *Adaptive Parameter Optimisation* framework for joint optimisation.

$\mathcal{D}_t$: Dataset for task t
$f_{\boldsymbol{\theta}}()$: Shared model with parameters $\theta \in \mathbb{R}^N$
$\mathbf{G}, \mathbf{P}$: Accumulating variable and global mask, initialised with zeros.
$\mathbf{H}^t$: Tracking variable for task t initialised with zeros.
η: Learning rate, μ: regularisation coefficient, k: threshold parameter, β: decay factor
$\theta_{prev} = \theta$
for $i \leftarrow 1 : N$ **do** ▷ N: Number of epochs
  for $t \leftarrow 1 : T$ **do** ▷ T: Number of Tasks
    $\mathcal{B} \leftarrow$ SAMPLE-BATCHES$(\mathcal{D}_t)$ ▷ batches
    for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{B}$ **do**
      $\mathbf{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})$ ▷ Gradients
      $\mathbf{G} = \beta \mathbf{G} + (1 - \beta)\mathbf{g}$ ▷ Gradient Accumulation
      $\tau = \text{mean}(\mathbf{G}) + k\ \text{std}(\mathbf{G})$ ▷ Threshold
      Initialise $\mathbf{M} \in \mathbb{R}^N$ with zeros
      for all element $\mathbf{M}_i$ of $\mathbf{M}$ **do**
        $\mathbf{M}_i = \begin{cases} 1, \text{ IF } |\mathbf{G}_i| > \tau \\ 0, \text{ OTHERWISE.} \end{cases}$
      end for
      $\mathbf{M}_{Protected} = (\mathbf{P} \wedge \neg \mathbf{H}^t) \wedge \mathbf{M}$
      $\mathbf{M}_{normal} = \mathbf{M} \wedge \neg\, \mathbf{M}_{Protected}$
      $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta(\mathbf{M}_{normal} \odot \mathbf{g})$
      $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta(\mathbf{M}_{Protective} \odot (\mathbf{g} + 2\mu(\boldsymbol{\theta} - \boldsymbol{\theta}_{prev})))$
      $\mathbf{H}^t = \mathbf{H}^t \ \vee \ \mathbf{M}_{normal}$
      $\mathbf{P} = \mathbf{P} \ \vee \ \mathbf{M}_{normal}$
    end for
  end for
  $\theta_{prev} = \theta$
end for

Although the shared model comprises both task-specific and shared layers, the task-specific layers are trained using standard procedures and are therefore omitted here for brevity. Our focus is on the shared layers, as their training across tasks presents a more intriguing aspect of joint optimisation.

4.2 Proposed adaptive parameter optimisation

Algorithm 1 documents the proposed APO framework for training the shared model $\boldsymbol{\theta}$ across T tasks. In each iteration, for each task t, the proposed framework performs the following operations:

- The current model state $\boldsymbol{\theta}$ is used to compute gradient $\mathbf{g}$ using a training batch $(\mathbf{x}, \mathbf{y})$ of task t as: $\mathbf{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}^t_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})$.
- These gradients for all parameters are accumulated

in $\mathbf{G}$ using an exponential moving average with decay factor β as: $\mathbf{G} = \beta\mathbf{G} + (1-\beta)\mathbf{g}$.

- APO utilises a dynamic thresholding mechanism over the accumulated gradients to select the sensitive gradients that can induce effective change in the training loss. This threshold τ is computed as: $\tau = \text{mean}(\mathbf{G}) + k\ \text{std}(\mathbf{G})$, where k determines the number of standard deviations above the mean used to set the threshold.

  Based on τ, we compute a mask $\mathbf{M}$ highlighting the sensitive parameters as: $\mathbf{M}_i = 1$ if $|\mathbf{G}_i| > \tau$, else 0, where $\mathbf{M}_i$ or $\mathbf{G}_i^t$ represent the ith element in corresponding tensors. Thresholding the accumulated gradients to identify sensitive parameters smooths out fluctuations over time, providing a more stable and reliable measure of parameter sensitivity that better captures long-term trends in learning dynamics.
- From $\mathbf{M}$, we identify the protected parameters, which are those associated with other tasks and must only be updated using proximal regularisation to minimise their deviation from their previous state: $\mathbf{M}_{Protected} = (\mathbf{P} \wedge \neg\mathbf{H}^t) \wedge \mathbf{M}$.

  The mask for remaining sensitive parameters, either associated with the current task t or unclaimed, can be obtained as: $\mathbf{M}_{normal} = \mathbf{M} \wedge \neg\, \mathbf{M}_{Protected}$.
- Using $\mathbf{M}_{normal}$ and gradients $\mathbf{g}$, $\boldsymbol{\theta}$ can be updated as:
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta(\mathbf{M}_{normal} \odot \mathbf{g}). \tag{2}$$
  Similarly, the protected parameters are updated using proximal regularisation that is inspired from FedProx (Li et al., 2020):
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta(\mathbf{M}_{Protective} \odot (\mathbf{g} + 2\mu(\boldsymbol{\theta} - \boldsymbol{\theta}_{prev}))). \tag{3}$$
  Here, $\boldsymbol{\theta}_{prev}$ is the parameter state in previous epoch, and μ controls the strength of regularisation.
- Finally, $\mathbf{H}^t$and $\mathbf{P}$ are updated to reflect the newly updated parameters by task t: $\mathbf{P} = \mathbf{P} \vee \mathbf{M}_{normal}$ and $\mathbf{H}^t = \mathbf{H}^t \vee \mathbf{M}_{normal}$.

Implementation details: In an APO epoch, all tasks sequentially update the shared model parameters, denoted as $\boldsymbol{\theta}$. The order in which tasks update the model is dynamic and is adjusted periodically based on each task's training loss, ensuring that APO prioritises tasks that require more attention. For simplicity, we previously described the shared model as a single trainable tensor, θ. However, in practice, deep learning models consist of multiple layers, each with its own set of trainable parameters or tensors. The operations described earlier, including the computation of dynamic thresholds and parameter updates, are applied to each tensor individually. This layer-wise and tensor-wise thresholding is crucial because the gradient magnitudes can vary significantly across different layers of the model.

4.3 Theoretical Insights

In this subsection, we present the theoretical foundations of the APO framework, demonstrating its capability to facilitate positive transfer among tasks while effectively minimising the impact of negative transfer.

Positive transfer: The first key property of the APO framework is its ability to promote positive transfer between tasks.

Theorem 1 (Positive Transfer)**.** *Let $\boldsymbol{\theta} \in \mathbb{R}^N$ be the shared model parameters optimised using the APO across tasks. Assume that the loss functions $L_t(\boldsymbol{\theta})$ for each task t satisfy the following conditions: convexity, Lipschitz continuity of gradients, bounded gradients, and bounded Hessians. Under these conditions, if the learning rate η_t and the scaling factors γ_i satisfy the condition:*
$$\eta_t\gamma_{\min} < \frac{2\rho}{H_{\max}}, \tag{4}$$
where $\gamma_{\min} = \min_i \gamma_i$ represents the minimum scaling factor applied by the protective update mechanism, $\rho > 0$ is a constant representing the minimum positive correlation between the gradients of consecutive tasks, and $H_{\max} > 0$ is the upper bound on the norm of the Hessian of the loss functions, then the unified APO update:
$$\Delta\boldsymbol{\theta}_t = -\eta_t\boldsymbol{\Gamma}\nabla L_t(\boldsymbol{\theta}), \tag{5}$$
where $\boldsymbol{\Gamma} = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_N)$, tends to align the gradient of task $t+1$ with that of task t and leads to a decrease in the loss function $L_{t+1}(\boldsymbol{\theta})$, thereby facilitating positive transfer between tasks.

Proof. Please refer to Section A of the supplementary document for a detailed proof. □

This theorem provides a theoretical foundation for how the APO framework enables positive transfer in joint optimisation. By utilising protective scaling factors γ_i in parameter updates guided by proximal regularisation (see proof of Theorem 1), APO effectively balances new learning with existing knowledge. The condition $\eta_t\gamma_{\min} < \frac{2\rho}{H_{\max}}$ ensures that even the most protected parameters contribute meaningfully to learning new tasks while preserving crucial information.

This result demonstrates that APO promotes gradient alignment between consecutive tasks, leading to a reduction in the loss function $L_{t+1}(\boldsymbol{\theta})$. By carefully tuning the learning rate η_t and the protective scaling factors γ_i which is controlled by μ in Equation 3, APO

leverages the positive gradient correlation ($\rho > 0$) between tasks to enhance overall performance.

Negative Transfer Avoidance: Complementing its ability to promote positive transfer, the APO framework also plays a critical role in minimising negative transfer.

Theorem 2 (Negative Transfer Avoidance)**.** *Assume that the loss functions $L_t(\boldsymbol{\theta})$ and $L_{t+1}(\boldsymbol{\theta})$ are convex, twice differentiable, and have Lipschitz continuous gradients. Furthermore, let the gradients and Hessians of the loss function be bounded such that the gradient norms satisfy $\|\nabla L_t(\boldsymbol{\theta})\| \leq G$ and $\|\nabla L_{t+1}(\boldsymbol{\theta})\| \leq G$, and the Hessian norm is bounded by $\|H_{t+1}\| \leq H_{\max}$.*

When the gradients of tasks t and $t+1$ are negatively aligned:

$$\nabla L_{t+1}(\boldsymbol{\theta})^\top \nabla L_t(\boldsymbol{\theta}) \leq -\rho G^2, \tag{6}$$

where $\rho > 0$ represents the degree of negative alignment, then using the unified update:

$$\Delta\boldsymbol{\theta}_t = -\eta_t \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}), \tag{7}$$

where $\boldsymbol{\Gamma} = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_N)$ with scaling factors $\gamma_i \in [\gamma_{\min}, 1]$ and $\gamma_{\max} = \max_i \gamma_i \leq 1$, ensures that the increase in the loss function $L_{t+1}(\boldsymbol{\theta})$ due to the update from task t is bounded by a small constant ϵ, provided that the learning rate η_t and the scaling factors γ_i satisfy:

$$\eta_t \gamma_{\max} \leq \min\left\{ \frac{\epsilon}{\rho G^2}, \quad \frac{\sqrt{2\epsilon}}{\sqrt{H_{\max} G}} \right\}. \tag{8}$$

Proof. Please refer to Section B of the supplementary document for a detailed proof. □

This theorem confirms that APO effectively minimises negative transfer by incorporating protective scaling factors γ_i into parameter updates, thereby limiting the magnitude of updates that could otherwise increase the loss of future tasks. The condition ensures that any increase in the loss function for task $t+1$ remains bounded, safeguarding the model's performance.

Together, Theorems 1 and 2 demonstrate that APO not only promotes positive transfer by aligning task gradients but also actively prevents negative transfer by modulating parameter updates. This dual capability of fostering synergy between tasks while mitigating conflicts highlights APO's robustness in multi-task learning, allowing for consistent performance across diverse tasks. The theoretical guarantees provide a strong foundation for APO's effectiveness in handling complex learning scenarios where task interdependencies are critical.

5 EXPERIMENTS

Datasets, tasks and models: The APO framework is evaluated on three clinical datasets from different modalities and a set of generic image datasets. These datasets are used to obtain four task categories, each jointly trained using the APO framework:

- Patient care tasks: Using the MIMIC-III dataset (Johnson et al., 2016), we predict in-hospital mortality (based on the first 48 hours of ICU stay), decompensation within the next 24 hours, and phenotypes using the entire ICU stays. Mortality and decompensation are binary classification tasks, while phenotyping involves assigning one or more conditions from 25 chronic and acute categories (Harutyunyan et al., 2019). Each ICU stay is represented as a time-series of 76 clinical measurements sampled hourly.
- Predictions using discharge notes: In these prediction tasks, MIMIC-III patient discharge summaries are utilised to predict mortality, readmission, sepsis, and phenotypes (spanning 10 broad disease categories) during ICU stays.
- Diagnosis using ECG signals: In these prediction tasks, one-minute ECG signals from the MIMIC-III waveform dataset (Moody, Benjamin et al., 2020) are processed into spectrograms to diagnose chronic kidney disorder, conduction disorders, coronary atherosclerosis, and hypertension.
- Image classification: CIFAR-10, STL-10, street view house number (SVHN) (Netzer et al., 2011), colorectal histology (Kather et al., 2016) and Malaria (Rajaraman et al., 2018) datasets are used for image classification tasks. CIFAR-10 and SVHN contain low-resolution images, while STL-10 has higher-resolution images. The colorectal histology and malaria datasets involve histopathology and single-cell images.

For the patient care tasks, we use a transformer-based time-series model. The ECG tasks are handled by a convolutional recurrent neural network. For text-based tasks, we employ a pre-trained TinyBERT model, and ResNet-50 is used for image classification. Detailed information about the models and data splits is provided in the supplementary document.

Comparative methods: We evaluate the proposed algorithm against several baselines, including Single Task Learning (STL), meta-learning approaches such as Reptile (Nichol and Schulman, 2018), MAML with implicit gradients (iMAML) (Rajeswaran et al., 2019), and kernel modulation-based meta-learning (KML) (Abdollahzadeh et al., 2021), as well as methods like Mixture-of-Experts (MoE) (Chen et al., 2023),

Table 1: Performance evaluation of the proposed APO framework on (a) patient care tasks and (b) discharge summaries tasks derived from the MIMIC-III dataset, using AUROC as the performance metric.

(a) Patient Care Tasks

Methods ↓	Phenotyping	Mortality	Decompensation	Average
STL	0.763 ± 0.016	0.851 ± 0.016	0.794 ± 0.013	0.803 ± 0.015
REPTILE	0.709 ± 0.006	0.818 ± 0.016	0.779 ± 0.022	0.769 ± 0.015
iMAML	0.723 ± 0.009	0.826 ± 0.015	0.785 ± 0.024	0.778 ± 0.016
LMC	0.771 ± 0.015	0.841 ± 0.015	0.791 ± 0.006	0.801 ± 0.012
MoE	0.748 ± 0.015	0.829 ± 0.024	0.815 ± 0.023	0.797 ± 0.021
GradNorm	0.726 ± 0.017	0.837 ± 0.015	0.787 ± 0.015	0.783 ± 0.016
KML	0.762 ± 0.019	0.841 ± 0.016	0.802 ± 0.021	0.801 ± 0.019
PCGrad	0.739 ± 0.024	0.831 ± 0.024	0.806 ± 0.022	0.792 ± 0.024
Proposed	0.778 ± 0.009	0.848 ± 0.007	0.823 ± 0.012	0.816 ± 0.010
Proposed+MoE	**0.792** ± 0.013	**0.852** ± 0.016	**0.832** ± 0.021	**0.825** ± 0.017

(b) Discharge Note Tasks

Methods ↓	Readmission	Phenotyping	Sepsis	Mortality	Average
STL	0.879 ± 0.013	0.591 ± 0.022	0.924 ± 0.019	0.755 ± 0.019	0.787 ± 0.018
REPTILE	0.877 ± 0.012	0.587 ± 0.022	0.921 ± 0.025	0.739 ± 0.026	0.781 ± 0.021
iMAML	0.880 ± 0.025	0.590 ± 0.024	0.925 ± 0.026	0.748 ± 0.019	0.786 ± 0.024
LMC	0.882 ± 0.024	0.594 ± 0.027	0.927 ± 0.011	**0.759** ± 0.016	0.790 ± 0.019
MoE	0.882 ± 0.024	0.595 ± 0.012	0.928 ± 0.026	0.739 ± 0.016	0.786 ± 0.020
GradNorm	0.876 ± 0.021	0.587 ± 0.021	0.929 ± 0.020	0.752 ± 0.020	0.786 ± 0.021
KML	0.871 ± 0.017	0.591 ± 0.023	0.935 ± 0.026	0.741 ± 0.020	0.785 ± 0.022
PCGrad	0.883 ± 0.014	0.594 ± 0.026	0.931 ± 0.022	0.748 ± 0.024	0.789 ± 0.022
Proposed	0.885 ± 0.026	0.602 ± 0.012	0.937 ± 0.019	0.748 ± 0.013	0.793 ± 0.018
Proposed+MoE	**0.897** ± 0.019	**0.608** ± 0.010	**0.941** ± 0.012	0.758 ± 0.015	**0.801** ± 0.014

Table 2: Performance of different methods on ECG-based prediction tasks. AUROC is used as the performance metric.

Methods ↓	Chronic Kidney Disorder	Conduction Disorder	Coronary Atherosclerosis	Hypertension	Average
STL	0.611 ± 0.012	0.722 ± 0.021	0.610 ± 0.010	0.610 ± 0.020	0.638 ± 0.016
REPTILE	0.767 ± 0.006	0.747 ± 0.017	0.676 ± 0.015	0.638 ± 0.017	0.707 ± 0.014
iMAML	0.771 ± 0.009	0.758 ± 0.018	0.648 ± 0.016	0.678 ± 0.020	0.714 ± 0.016
LMC	0.735 ± 0.017	0.741 ± 0.024	0.636 ± 0.006	0.631 ± 0.020	0.686 ± 0.017
MoE	0.749 ± 0.011	0.759 ± 0.015	0.665 ± 0.018	0.647 ± 0.020	0.705 ± 0.016
GradNorm	0.791 ± 0.022	0.775 ± 0.020	0.695 ± 0.005	0.635 ± 0.021	0.724 ± 0.017
KML	0.775 ± 0.019	0.749 ± 0.017	0.634 ± 0.015	0.642 ± 0.024	0.7 ± 0.019
PCGrad	0.760 ± 0.020	0.749 ± 0.010	0.659 ± 0.009	0.729 ± 0.006	0.724 ± 0.011
Proposed	0.785 ± 0.009	0.795 ± 0.014	**0.699** ± 0.007	0.711 ± 0.024	0.747 ± 0.014
Proposed+MoE	**0.794** ± 0.018	**0.832** ± 0.023	0.691 ± 0.018	0.693 ± 0.009	**0.752** ± 0.017

Linear Mode Connectivity (LMC)-based multi-tasking (Mirzadeh et al., 2021), GradNorm (Chen et al., 2018), and Projecting Conflicting Gradients (PCGrad) (Yu et al., 2020). Apart from that, we also add a new baseline by combining APO and MoE (APO+MoE). Unlike other baselines that focus on optimization or gradient manipulation, MoE introduces a structural change via a gating mechanism that enables task-specific representations. This complements APO, which optimizes parameter updates across tasks, making their combination (APO+MoE) a compelling baseline.

Parameter settings for all methods are tuned to achieve the best validation performance, as detailed in Section D of the appendix.

6 RESULTS & DISCUSSION

6.1 Clinical prediction tasks

Patient care tasks: Table 1(a) compares the performance of the proposed APO framework against other methods on three patient-care tasks. Standard information-sharing techniques like Reptile and iMAML demonstrate lower average performance compared to single-task learning (STL), indicating a negative transfer effect. The Mixture-of-Experts (MoE) framework narrows the performance gap but still lags slightly behind STL. In contrast, the proposed APO framework demonstrates a relative improvement of approximately 2% over the best baseline, STL, highlighting its superior ability to leverage shared information across tasks while avoiding negative transfer. The combination of APO with MoE further enhances performance, achieving an additional 1% improvement over APO alone, solidifying its position as the best-performing approach. Moreover, this combined approach consistently outperforms STL across all tasks, underscoring its robustness.

Discharge summaries tasks: The performance of different methods on these tasks is documented in Table 1(b). The analysis indicates that the baseline methods deliver relatively comparable performance, with LMC and PCGrad slightly outperforming STL, achieving modest relative improvements of only 0.4% and 0.3%, respectively. In contrast, the proposed APO framework demonstrates a substantial performance boost over STL across all tasks except mortality prediction, resulting in a consistent average improvement over STL and all other baselines. Furthermore, the combination of APO with MoE leads to a notable additional gain, yielding a 1% relative improvement over APO alone, solidifying its position as the most effective approach among all methods

ECG-based prediction tasks: The performance on ECG-based diagnosis tasks is presented in Table 2. Standard baselines like Reptile and iMAML improve over single-task learning (STL) by approximately 11% and 12%, respectively, leveraging information-sharing mechanisms effectively. However, methods like LMC and MoE show mixed results, with sporadic task-specific improvements but inconsistent overall gains. In contrast, the proposed APO framework delivers a substantial performance boost, achieving an average relative gain of 17% over STL, with notable strength in the coronary atherosclerosis task.

While integrating APO with MoE yields slight improvements (0.5% over APO alone), its performance varies across tasks. APO+MoE excels in chronic kidney disease (CKD) and conduction disorder predictions but falls behind APO in coronary atherosclerosis and hypertension. This divergence arises because MoEs gating mechanism, though effective in task allocation, struggles when task differences are

Table 3: Performance of different methods across image classification tasks.

Tasks ↓	Methods									
	STL	REPTILE	iMAML	LMC	MoE	GradNorm	KML	PCGrad	APO	APO+MoE
CIFAR-10	83.40 ± 0.18	82.13 ± 0.14	82.30 ± 0.12	83.67 ± 0.18	83.88 ± 0.15	82.95 ± 0.12	83.67 ± 0.16	83.71 ± 0.13	84.05 ± 0.11	$\mathbf{84.14} \pm 0.16$
SVHN	89.22 ± 0.14	88.13 ± 0.12	88.28 ± 0.08	88.46 ± 0.14	88.96 ± 0.19	88.64 ± 0.20	89.14 ± 0.19	88.79 ± 0.16	89.32 ± 0.14	$\mathbf{89.39} \pm 0.17$
STL-10	41.66 ± 0.11	41.88 ± 0.08	42.28 ± 0.08	43.30 ± 0.12	44.01 ± 0.15	43.40 ± 0.12	43.68 ± 0.19	43.85 ± 0.18	44.73 ± 0.18	$\mathbf{44.97} \pm 0.16$
Malaria	97.13 ± 0.14	96.01 ± 0.13	96.27 ± 0.17	96.54 ± 0.17	97.05 ± 0.17	96.70 ± 0.19	96.38 ± 0.10	96.79 ± 0.13	$\mathbf{97.19} \pm 0.13$	97.18 ± 0.13
Colorectal	84.34 ± 0.09	82.37 ± 0.19	83.59 ± 0.19	84.96 ± 0.09	85.82 ± 0.18	84.99 ± 0.18	85.24 ± 0.20	84.85 ± 0.12	85.42 ± 0.15	$\mathbf{85.74} \pm 0.16$
Average	79.15 ± 0.13	78.10 ± 0.13	78.54 ± 0.13	79.39 ± 0.14	79.94 ± 0.17	79.34 ± 0.16	79.62 ± 0.17	79.60 ± 0.14	80.14 ± 0.14	$\mathbf{80.28} \pm 0.16$

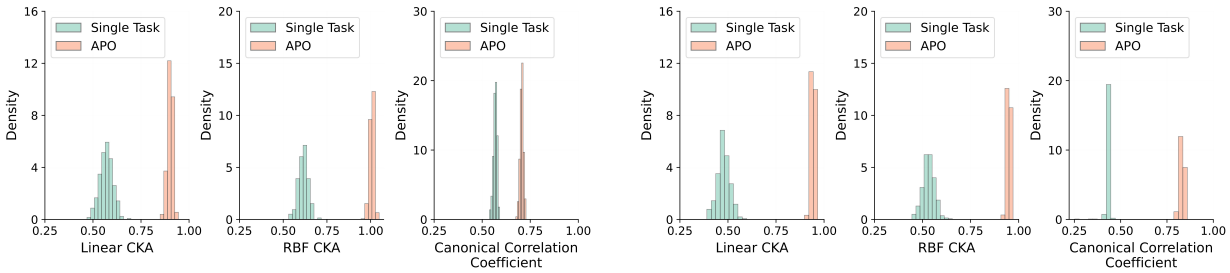


Figure 3: Central Kernel Alignment (CKA) and Canonical Correlation Coefficient-based similarity between representations are shown for ***(left)*** mortality and phenotype examples, and ***(right)*** decompensation and mortality samples, computed from the single-task models and the APO model.

subtle. Conditions like CKD and conduction disorders exhibit distinct spectral patternse.g., CKD-induced hyperkalemia leads to peaked T waves and widened QRS complexes, while conduction disorders show prolonged QRS durationsallowing MoE to allocate experts effectively. In contrast, hypertension and coronary atherosclerosis lack clear spectral signatures, with spectral changes being more subtle and distributed. This reduces the effectiveness of MoEs expert allocation, explaining its inconsistent benefits in APO+MoE.

6.2 Image classification tasks

Table 3 presents the performance of different methods across image classification tasks. Consistent with earlier observations, baseline methods like Reptile and iMAML deliver moderate improvements but generally fall short of single-task learning (STL), indicating limited effectiveness in their information-sharing strategies. LMC and MoE show more promising results, achieving relative gains of up to 1% over STL on average, suggesting a better ability to exploit shared knowledge across tasks.

The proposed APO framework continues to demonstrate a significant advantage, outperforming STL by an average of approximately 1.25% across tasks, with notable improvements in CIFAR-10, SVHN, and Colorectal classification. Although APO achieves the best performance individually in the Malaria classification task, the combination of APO with MoE offers only a slight relative gain of about 0.2% over APO alone. Despite this minor incremental gain, APO+MoE consistently achieves the highest performance across most tasks, including CIFAR-10, SVHN, and STL-10, highlighting its robust ability to generalise effectively. This result aligns with our earlier findings, where APO's combination with MoE yielded varying improvements, reinforcing its potential as a versatile and powerful approach across different application domains.

6.3 Information sharing & task-specific characteristics

We analyse the penultimate layer embeddings (outputs from the shared layers) generated by the model trained on MIMIC-III patient care tasks using APO and compare these embeddings to those from single-task models. Figure 3 shows the similarity distributions between embeddings generated for phenotyping and mortality examples using the APO model compared to the respective single-task models. It also displays the similarity distribution between embeddings for mortality and decompensation samples. We measure these similarities using central kernel alignments (CKA) with both linear and RBF (radial basis function) kernels, along with canonical correlation analysis (CCA) (Kornblith et al., 2019). This analysis demonstrates that the APO produce embeddings that share significant similarities, indicating enhanced information transfer between tasks. However, the embeddings are not identical (similarity equal to 1), preserving the essential task-specific characteristics that are often lost in traditional multi-task learning or fixed global representa-

Table 4: Comparison of dynamic vs. fixed task ordering for patient-care tasks (MIMIC-III).

Task	Phenotyping	Mortality	Decompensation	Average
Dynamic	**0.792** (0.013)	**0.852** (0.016)	**0.832** (0.021)	**0.825** (0.017)
Fixed	0.783 (0.016)	**0.852** (0.014)	0.827 (0.018)	0.821 (0.016)

tion methods.

Table 5: Comparison of dynamic vs. fixed task ordering for discharge notes tasks.

Task	Readmission	Phenotyping	Sepsis	Mortality	Average
Dynamic	**0.897** (0.019)	**0.608** (0.010)	0.941 (0.012)	**0.758** (0.015)	**0.801** (0.014)
Fixed	0.896 (0.017)	0.587 (0.012)	**0.942** (0.011)	0.751 (0.013)	0.794 (0.013)

6.4 Impact of dynamic task ordering

To evaluate the impact of dynamic task ordering in APO, we conducted an ablation study by replacing the dynamic scheduling mechanism with a fixed task sequence maintained throughout training. In the default APO implementation, task updates follow a dynamic order, updated every 5 epochs, with higher-loss tasks prioritized to enhance training efficiency. This study examines patient-care and discharge notes tasks, with results presented in Tables 4 and 5.

The results indicate that using a fixed task sequence leads to a slight performance decline across both task groups. Notably, tasks appearing later in the fixed ordersuch as Phenotyping in patient-care tasks and Mortality/Phenotyping in discharge notes tasksexperience a more pronounced drop. This decline is primarily due to resource starvation, where later tasks have fewer parameters for task-specific adaptation, and diminished gradient relevance, as shared parameters become increasingly tailored to earlier tasks, making updates for later tasks less effective. Dynamic ordering mitigates these issues by prioritizing higher-loss tasks, ensuring better resource allocation, and reducing negative transfer.

7 CONCLUSION & LIMITATIONS

This paper presents the Adaptive Parameter Optimisation (APO) framework, which enhances information sharing across related tasks while mitigating negative transfer. APO selectively updates sensitive parameters and uses a protective update mechanism to preserve knowledge from previous tasks, promoting controlled lazy behaviour. This approach is particularly relevant to healthcare use cases where data is scarce and tasks are interrelated. Experiments on healthcare and generic image datasets show that APO outperforms standard single-task models as well as comparative joint optimisation strategies. Theoretical analyses, including proofs on positive transfer and negative transfer avoidance, validate APO's effectiveness in aligning gradients and minimising negative transfer.

However, an important limitation of the proposed approach is its scalability concerning the number of tasks accommodated within a shared model. The modelling complexity, number of parameters, and diversity of task types determine the maximum number of tasks that can effectively share a model without performance degradation. As the shared parameters become saturated, APO's ability to adapt the shared layers to new tasks diminishes, effectively turning the proposed method into a soft-parameter sharing paradigm due to the protective update mechanism. Future work could explore strategies to mitigate this limitation, such as dynamic allocation of parameters or hierarchical modelling techniques that allow for efficient sharing across a larger number of tasks.

Code

The implementation of the proposed method is available at https://github.com/AnshThakur/Adaptive_Parameter_Optimisation.

Acknowledgements

David A. Clifton is supported by the Pandemic Sciences Institute at the University of Oxford; the National Institute for Health Research (NIHR) Oxford Biomedical Research Center (BRC); an NIHR Research Professorship; a Royal Academy of Engineering Research Chair; the Wellcome Trust; the UKRI; and the InnoHK Hong Kong Center for Center for Cerebrocardiovascular Engineering (COCHE).

References

M. Abdollahzadeh, T. Malekzadeh, and N.-M. M. Cheung. Revisit multimodal meta-learning through the lens of multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252, 2019.

S. Allenspach, J. A. Hiss, and G. Schneider. Neural multi-task learning in drug design. *Nature Machine Intelligence*, 6(2):124–137, 2024.

D. Benavides-Prado and P. Riddle. A theory for knowledge transfer in continual learning. In *Conference on Lifelong Learning Agents*, pages 647–660. PMLR, 2022.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

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.

Z. Chen, Y. Shen, M. Ding, Z. Chen, H. Zhao, E. G. Learned-Miller, and C. Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837, 2023.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(96):1–18, 2019.

A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

J. N. Kather, C.-A. Weis, F. Bianconi, S. M. Melchers, L. R. Schad, T. Gaiser, A. Marx, and F. G. Z"ollner. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports*, 6:27988, 2016.

A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in Neural Information Processing Systems*, 31, 2018.

X. Liu, L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern, et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health*, 1(6):e271–e297, 2019.

J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018.

S. I. Mirzadeh, M. Farajtabar, D. Gorur, R. Pascanu, and H. Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Fmg_fQYUejf.

I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

Moody, Benjamin, Moody, George, Villarroel, Mauricio, Clifford, Gari D., and Silva, Ikaro. Mimic-iii waveform database (version 1.0). https://physionet.org/content/mimic3wdb/1.0/, 2020. Accessed: 2024-07-15.

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.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger, and G. R. Thoma. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6: e4568, 2018.

A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, 1(1): 1–10, 2018.

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

A. Thakur, P. Sharma, and D. A. Clifton. Dynamic neural graphs based federated reptile for semi-supervised multi-tasking in healthcare applications. *IEEE Journal of Biomedical and Health Informatics*, 2021.

A. Thakur, V. Abrol, P. Sharma, T. Zhu, and D. A. Clifton. Incremental trainable parameter selection in deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation. *Advances in Neural Information Processing Systems*, 32, 2019.

H. Wang, H. Zhao, and B. Li. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In *International Conference on Machine Learning*, pages 10991–11002. PMLR, 2021.

J. Wilkinson, K. F. Arnold, E. J. Murray, M. van Smeden, K. Carr, R. Sippy, M. de Kamps, A. Beam, S. Konigorski, C. Lippert, et al. Time to reality check the promises of machine learning-powered precision medicine. *The Lancet Digital Health*, 2(12): e677–e680, 2020.

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A THEOREM 1: POSITIVE TRANSFER

A.1 Assumptions

Let $\boldsymbol{\theta} \in \mathbb{R}^N$ be the shared model parameters. For each task t, let $L_t(\boldsymbol{\theta})$ be its loss function. We make the following assumptions:

- Convexity and Smoothness: Each $L_t(\boldsymbol{\theta})$ is convex and twice differentiable. Also, the gradients are Lipschitz continuous with constant $C_L > 0$:

$$\|\nabla L_t(\boldsymbol{\theta}) - \nabla L_t(\boldsymbol{\theta'})\| \leq C_L \|\boldsymbol{\theta} - \boldsymbol{\theta'}\|, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta'} \in \mathbb{R}^N. \tag{9}$$

- Bounded Gradients:

$$\|\nabla L_t(\boldsymbol{\theta})\| \leq G, \quad \forall t. \tag{10}$$

- Bounded Hessians:

$$\|H_t\| \leq H_{\max}, \quad \forall t, \tag{11}$$

  where $H_t = \nabla^2 L_t(\boldsymbol{\theta})$.

- Positive Gradient Correlation (if present):

$$\nabla L_t(\boldsymbol{\theta})^\top \nabla L_{t+1}(\boldsymbol{\theta}) \geq \rho G^2, \quad \rho > 0. \tag{12}$$

- Negative Gradient Correlation (if present):

$$\nabla L_t(\boldsymbol{\theta})^\top \nabla L_{t+1}(\boldsymbol{\theta}) \leq -\rho G^2, \quad \rho > 0. \tag{13}$$

A.2 Unified update and protective scaling factor γ

In both theorems, we use a unified parameter update for both protective and normal parameters. Also, we introduce a protective scaling factor γ_i that scales down the impact of gradient updates for protective parameters. In this subsection, we study the relationship between unified updates, γ_i and regularisation constant μ in protective updates.

In our method, the protective update mechanism introduces a regularisation term in the loss function:

$$L_t^{\text{total}}(\boldsymbol{\theta}) = L_t(\boldsymbol{\theta}) + \mu \sum_i M_i (\theta_i - \theta_{i,\text{prev}})^2, \tag{14}$$

where $L_t(\boldsymbol{\theta})$ is the loss function for task t, μ is the regularisation coefficient controlling the strength of the protection, M_i is the protective mask for parameter θ_i ($M_i = 1$ if protected, $M_i = 0$ otherwise), and $\theta_{i,\text{prev}}$ is the previous value of parameter θ_i.

The gradient of the total loss with respect to θ_i is:

$$\nabla_{\theta_i} L_t^{\text{total}}(\boldsymbol{\theta}) = \nabla_{\theta_i} L_t(\boldsymbol{\theta}) + 2\mu M_i (\theta_i - \theta_{i,\text{prev}}). \tag{15}$$

The parameter update is then:

$$\Delta\theta_i = -\eta_t \left(\nabla_{\theta_i} L_t(\boldsymbol{\theta}) + 2\mu M_i(\theta_i - \theta_{i,\text{prev}}) \right). \tag{16}$$

We aim to align this with the unified update in the theorem:

$$\Delta\theta_i = -\eta_t \gamma_i \nabla_{\theta_i} L_t(\boldsymbol{\theta}), \tag{17}$$

where γ_i is the scaling factor for parameter θ_i. To find γ_i, we set the two expressions for $\Delta\theta_i$ equal to each other:

$$-\eta_t \gamma_i \nabla_{\theta_i} L_t(\boldsymbol{\theta}) = -\eta_t \left(\nabla_{\theta_i} L_t(\boldsymbol{\theta}) + 2\mu M_i(\theta_i - \theta_{i,\text{prev}}) \right). \tag{18}$$

Dividing both sides by $-\eta_t$:

$$\gamma_i \nabla_{\theta_i} L_t(\boldsymbol{\theta}) = \nabla_{\theta_i} L_t(\boldsymbol{\theta}) + 2\mu M_i(\theta_i - \theta_{i,\text{prev}}). \tag{19}$$

$$\gamma_i = 1 + \frac{2\mu M_i(\theta_i - \theta_{i,\text{prev}})}{\nabla_{\theta_i} L_t(\boldsymbol{\theta})}. \tag{20}$$

However, this expression depends on $\theta_i - \theta_{i,\text{prev}}$ and $\nabla_{\theta_i} L_t(\boldsymbol{\theta})$, which are dynamic quantities. To simplify, we consider that the change in parameter is small, so $\theta_i - \theta_{i,\text{prev}} \approx \Delta\theta_i$. Substituting $\Delta\theta_i$ into the expression:

$$\gamma_i = 1 + \frac{2\mu M_i \Delta\theta_i}{\nabla_{\theta_i} L_t(\boldsymbol{\theta})}. \tag{21}$$

But $\Delta\theta_i = -\eta_t \gamma_i \nabla_{\theta_i} L_t(\boldsymbol{\theta})$, so:

$$\gamma_i = 1 + \frac{2\mu M_i(-\eta_t \gamma_i \nabla_{\theta_i} L_t(\boldsymbol{\theta}))}{\nabla_{\theta_i} L_t(\boldsymbol{\theta})} = 1 - 2\mu\eta_t M_i \gamma_i. \tag{22}$$

Solving for γ_i:

$$\gamma_i = 1 - 2\mu\eta_t M_i \gamma_i. \tag{23}$$

Bringing like terms together:

$$\gamma_i + 2\mu\eta_t M_i \gamma_i = 1 \implies \gamma_i(1 + 2\mu\eta_t M_i) = 1. \tag{24}$$

Therefore, the scaling factor γ_i is:

$$\boxed{\gamma_i = \frac{1}{1 + 2\mu\eta_t M_i}.} \tag{25}$$

For protected parameters ($M_i = 1$), so $\gamma_i = \frac{1}{1+2\mu\eta_t} < 1$, which means the update is scaled down, providing protection.

For task-specific parameters, we have no proximal regularisation, so $\mu = 0$ and hence, $\gamma_i = \frac{1}{1+0} = 1$, which means the update reduces to the standard gradient descent update.

Hence, based on the γ, we have arrived at an unified update.

A.3 Proof of Theorem 1 (Positive Transfer Theorem)

We will show that the unified update $\Delta\boldsymbol{\theta}_t$ leads to positive transfer by decreasing $L_{t+1}(\boldsymbol{\theta})$ and improving gradient alignment.

As discussed earlier, the update for all parameters is:

$$\Delta\boldsymbol{\theta}_t = -\eta_t \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}). \tag{26}$$

After applying this task t update, the gradient of task $t+1$ can be approximated using first-order Taylor expansion:

$$\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx \nabla L_{t+1}(\boldsymbol{\theta}) + H_{t+1}\Delta\boldsymbol{\theta}_t. \tag{27}$$

Substituting $\Delta\boldsymbol{\theta}_t$ from Equation 26:

$$\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx \nabla L_{t+1}(\boldsymbol{\theta}) - \eta_t H_{t+1} \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}). \tag{28}$$

Now, we evaluate the alignment between approximated gradient of task $t+1$ with $\nabla L_t(\boldsymbol{\theta})$:

$$\left(\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t)\right)^\top \nabla L_t(\boldsymbol{\theta}) \approx \nabla L_{t+1}(\boldsymbol{\theta})^\top \nabla L_t(\boldsymbol{\theta}) - \eta_t \nabla L_t(\boldsymbol{\theta})^\top H_{t+1} \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}). \tag{29}$$

From the positive gradient correlation assumption, we have:

$$\nabla L_{t+1}(\boldsymbol{\theta})^\top \nabla L_t(\boldsymbol{\theta}) \geq \rho G^2. \tag{30}$$

Following the bounder Hessians and Gradient assumptions, we obtain:

$$\nabla L_t(\boldsymbol{\theta})^\top H_{t+1} \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}) \leq \|H_{t+1}\| \|\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\| \|\nabla L_t(\boldsymbol{\theta})\| \leq H_{\max} \gamma_{\max} G^2, \tag{31}$$

where $\gamma_{max} = 1$. Thus, we arrive at:

$$\left(\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t)\right)^\top \nabla L_t(\boldsymbol{\theta}) \geq \rho G^2 - \eta_t H_{\max} G^2. \tag{32}$$

For positive alignment,

$$\rho G^2 - \eta_t H_{\max} G^2 > 0 \implies \eta_t < \frac{\rho}{H_{\max}}. \tag{33}$$

Now we evaluate the decrease in loss for task $t+1$. Using a second-order Taylor expansion:

$$L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx L_{t+1}(\boldsymbol{\theta}) + \nabla L_{t+1}(\boldsymbol{\theta})^\top \Delta\boldsymbol{\theta}_t + \frac{1}{2} \Delta\boldsymbol{\theta}_t^\top H_{t+1} \Delta\boldsymbol{\theta}_t. \tag{34}$$

Substituting $\Delta\boldsymbol{\theta}_t$:

$$L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx L_{t+1}(\boldsymbol{\theta}) - \eta_t \nabla L_{t+1}(\boldsymbol{\theta})^\top \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}) + \frac{1}{2} \eta_t^2 \left(\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\right)^\top H_{t+1} \left(\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\right). \tag{35}$$

Again, using the following bounds:

- Since $\gamma_i \geq \gamma_{\min}$,

$$\nabla L_{t+1}(\boldsymbol{\theta})^\top \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}) \geq \gamma_{\min} \rho G^2. \tag{36}$$

- The second-order term satisfies:

$$(\mathbf{\Gamma}\nabla L_t(\boldsymbol{\theta}))^\top H_{t+1} (\mathbf{\Gamma}\nabla L_t(\boldsymbol{\theta})) \leq H_{\max}\gamma_{\max}^2 G^2 = H_{\max} G^2. \tag{37}$$

Hence, we can derive the change in loss from Equation 35 as:

$$\Delta L_{t+1} = L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) - L_{t+1}(\boldsymbol{\theta}) \leq -\eta_t \gamma_{\min} \rho G^2 + \frac{1}{2}\eta_t^2 H_{\max} G^2. \tag{38}$$

For $\Delta L_{t+1} < 0$, we require:

$$-\eta_t \gamma_{\min} \rho + \frac{1}{2}\eta_t^2 H_{\max} < 0. \tag{39}$$

Solving the inequality:

$$\boxed{\eta_t \gamma_{\min} < \frac{2\rho}{H_{\max}}.} \tag{40}$$

Also, when this inequality is satisfied, the inequality in Equation 25 is automatically satisfied.

Hence, we conclude, under $\eta_t \gamma_{\min} < \frac{2\rho}{H_{\max}}$, the loss $L_{t+1}(\boldsymbol{\theta})$ decreases and gradient remain positively aligned.

B THEOREM 2 PROOF: NEGATIVE TRANSFER AVOIDANCE

Assumptions mentioned in Section A.1 also apply to Theorem 2 as well.

We are given that the gradients of tasks t and $t+1$ are negatively aligned:

$$\nabla L_{t+1}(\boldsymbol{\theta})^\top \nabla L_t(\boldsymbol{\theta}) \leq -\rho G^2. \tag{41}$$

This means that updating the parameters in the direction of $\nabla L_t(\boldsymbol{\theta})$ could potentially increase the loss for task $t+1$.

As earlier, we use the unified parameter update:

$$\Delta\boldsymbol{\theta}_t = -\eta_t \mathbf{\Gamma} \nabla L_t(\boldsymbol{\theta}), \tag{42}$$

where $\eta_t > 0$ is the learning rate, $\mathbf{\Gamma}$ is a diagonal matrix with scaling factors $\gamma_i \in [\gamma_{\min}, 1]$, and $\gamma_{\max} = \max_i \gamma_i \leq 1$. The protective scaling factors γ_i reduce the magnitude of updates to mitigate negative transfer.

Using a first-order Taylor expansion, the gradient of task $t+1$ after the update is approximated as:

$$\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx \nabla L_{t+1}(\boldsymbol{\theta}) + H_{t+1}\Delta\boldsymbol{\theta}_t. \tag{43}$$

Substituting the update in this equation, we get:

$$\nabla L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) \approx \nabla L_{t+1}(\boldsymbol{\theta}) - \eta_t H_{t+1} \mathbf{\Gamma} \nabla L_t(\boldsymbol{\theta}). \tag{44}$$

Now, we compute the change in loss function L_{t+1} using second-order Taylor expansion:

$$\Delta L_{t+1} = L_{t+1}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_t) - L_{t+1}(\boldsymbol{\theta}) \approx \nabla L_{t+1}(\boldsymbol{\theta})^\top \Delta\boldsymbol{\theta}_t + \frac{1}{2}\Delta\boldsymbol{\theta}_t^\top H_{t+1} \Delta\boldsymbol{\theta}_t. \tag{45}$$

Substituting $\Delta\boldsymbol{\theta}_t$:

$$\Delta L_{t+1} \approx -\eta_t \nabla L_{t+1}(\boldsymbol{\theta})^\top \mathbf{\Gamma} \nabla L_t(\boldsymbol{\theta}) + \frac{1}{2}\eta_t^2 (\mathbf{\Gamma}\nabla L_t(\boldsymbol{\theta}))^\top H_{t+1} (\mathbf{\Gamma}\nabla L_t(\boldsymbol{\theta})). \tag{46}$$

Now, we aim to bound the first term : $-\eta_t \nabla L_{t+1}(\boldsymbol{\theta})^\top \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})$. Using the negative alignment condition and the fact that $\gamma_i \leq \gamma_{\max}$, we have:

$$\nabla L_{t+1}(\boldsymbol{\theta})^\top \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}) \leq \gamma_{\max} \nabla L_{t+1}(\boldsymbol{\theta})^\top \nabla L_t(\boldsymbol{\theta}) \leq -\gamma_{\max} \rho G^2. \tag{47}$$

Therefore, the first-order term is bounded above by:

$$-\eta_t \nabla L_{t+1}(\boldsymbol{\theta})^\top \boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta}) \leq \eta_t \gamma_{\max} \rho G^2. \tag{48}$$

For second-order term, we use boundedness of the Hessian and gradients, and $\gamma_i \leq \gamma_{\max}$:

$$\left(\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\right)^\top H_{t+1} \left(\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\right) \leq \|H_{t+1}\| \|\boldsymbol{\Gamma} \nabla L_t(\boldsymbol{\theta})\|^2 \leq H_{\max} \left(\gamma_{\max} G\right)^2 = H_{\max} \gamma_{\max}^2 G^2. \tag{49}$$

Therefore, the second-order term is bounded above by $\frac{1}{2} \eta_t^2 H_{\max} \gamma_{\max}^2 G^2$.

Finally, from Equation 38 and the above bounds, the change in loss is given by:

$$\Delta L_{t+1} \leq \eta_t \gamma_{\max} \rho G^2 + \frac{1}{2} \eta_t^2 H_{\max} \gamma_{\max}^2 G^2. \tag{50}$$

To ensure $\Delta L_{t+1} \leq \epsilon$, we require:

1. From the first-order term:
$$\eta_t \gamma_{\max} \rho G^2 \leq \epsilon \quad \Longrightarrow \quad \eta_t \gamma_{\max} \leq \frac{\epsilon}{\rho G^2}. \tag{51}$$

2. From the second-order term:
$$\frac{1}{2} \eta_t^2 H_{\max} \gamma_{\max}^2 G^2 \leq \epsilon \quad \Longrightarrow \quad \eta_t \gamma_{\max} \leq \frac{\sqrt{2\epsilon}}{\sqrt{H_{\max}} G}. \tag{52}$$

Therefore, combining both conditions:

$$\boxed{\eta_t \gamma_{\max} \leq \min \left\{ \frac{\epsilon}{\rho G^2}, \quad \frac{\sqrt{2\epsilon}}{\sqrt{H_{\max}} G} \right\}.} \tag{53}$$

By appropriately choosing the learning rate η_t and the scaling factors γ_i such that $\gamma_{\max} \leq 1$ and the above condition on $\eta_t \gamma_{\max}$ holds, we can ensure that the increase in the loss function $L_{t+1}(\boldsymbol{\theta})$ due to the update from task t is bounded by ϵ.

This effectively minimises the impact of negative transfer, as the protective scaling factors γ_i reduce the update magnitudes for parameters, preventing significant increases in the loss of task $t+1$.

C DATASET DETAILS & MODELS

DATASETS: Table A1 documents the information about the datasets, tasks and train-test splits used for the experimentation.

C.1 Pre-processing ECG data

MIMIC-III waveforms contain the ECG signals having a sampling rate of 256 Hz. A short-term Fourier transform is performed on each signal using a segment size of 64 samples and an overlap of 32 samples to obtain spectrograms (time-frequency representation). Spectrograms are normalised and converted to a log scale to obtain the final representation of an ECG signal.

Table A1: Characteristics of datasets used for experimentation.

Task Grouping	Datasets	Tasks	# Train, Test & Validation Examples
Patient Care Tasks (Time-series)	MIMIC-III	Mortality Prediction	14698, 3222 & 3236
		Decompensation Prediction	2388414, 520000 & 523208
		Phenotyping	29280, 6371 & 6281
Diagnosis using ECG signals	MIMIC-III Waveforms	Chronic Kidney Disorder	382, 128 & 128
		Conduction Disorder	429, 143 & 144
		Coronary Atherosclerosis	1015, 339 & 339
		Hypertension	400, 134 & 134
Predictions using Discharge Notes	MIMIC-III	Readmission	31.5K, 27K & 31.5K
		Phenotyping	56K, 48K & 56K
		Sepsis	35K, 30K & 35K
		Mortality	31.5K, 27K & 31.5K
Images	CIFAR-10	Classification	37.5K, 12.5K & 10K
	SVHN-10	Cropped Digit Classification	58.7K, 14.7K & 26K
	STL-10	Classification	2.8K, 1K & 1.2K
	Malaria	Malaria Detection using Cell Images	16.5K, 5.5K & 5.5K
	Colorectal histology	8-category texture classification	2.8K, 1K & 1.2K

C.2 Model Architectures

Patient Care Tasks: A transformer-based model is used for these tasks. The shared layers of model include a 256-node input embedding layer, followed by a Transformer encoder having 2 encoding layers each with 8 attention heads, and finally a linear layer with 128 nodes and the relU activation. The output of this linear layer is fed to task-specific layers for predictions. The task-specific output layers for mortality and decompensation consist of 1 node followed by sigmoid layer. Similarly, for phenotyping, the output layer has 25 nodes followed by sigmoid activation.

Discharge Notes Tasks: The model used for these tasks employs a shared TinyBERT layer to extract compact 312-d representations from input text, followed by a shared fully connected (FC) layer with 128 nodes that reduces the dimensionality of these features. The output of this shared FC layer are fed into task-specific linear layers then generate predictions for each task.

ECG prediction tasks: Convolutional recurrent neural networks (CRNN) having the following architecture are used for these tasks:

Conv layer($5 \times 5, 32$ filters) $\rightarrow$ Pooling(2×1) $\rightarrow$ Conv layer($5 \times 5, 64$ filters) $\rightarrow$
Pooling(4×1) $\rightarrow$ Conv layer($5 \times 5, 128$ filters) $\rightarrow$ Pooling(4×1) $\rightarrow$

Table A2: Hyperparameters used in APO for all task groupings.

Task grouping	Batch-size	Optimiser	Learning Rate	μ	k	β	Epochs
Patient Care Tasks	64	SGD	0.01	1	0.5	0.99	300
Discharge Notes	512		0.01		0.5		100
ECG Tasks	64		0.001		0.75		1500
Image Classification	128		0.001		0.5		200

Conv layer($5 \times 5, 256$ filters) $\rightarrow$ Pooling(4×1) $\rightarrow$ LSTM with 32 recurrent units $\rightarrow$ Dropout with 0.5 rate $\rightarrow$ Dense Layer with 1 node $\rightarrow$ Sigmoid activation.

The last linear layer is task-specific output layer while all other layers are shared across tasks. Again, binary cross-entropy is used as the loss function for all the tasks.

Image tasks: The shared layers of the model consists of a Feature Extraction module consisting of pre-trained Resnet-50 followed by linear layer with 256 nodes and relu activation. These shared layers are followed by task-specific output layers having n nodes. Here, n is dependent on tasks.

D PARAMETER SETTINGS IN APO

Table A2 document the major hyperparameters used in the proposed APO framework. All the method-specific parameters are tuned to provide the best average performance on the validation examples.

The learning rate mentioned here is for the normal updates. For protective parameter updates, we use dampen the learning rate by a factor of 100 (dividing learning rate by 100) in for every task. This dampening, along with regularisation provide finer control on scaling factors mentioned in theorems. The search space chosen for tuning this dampening factor is {10,50,100,500,1000}.

The proximal regularisation coefficient μ is chosen from the search space $\{0.01, 0.1, 0.5, 1, 2, 5\}$.

The k in dynamic threshold computation is chosen from the pre-determined search space $\{0.1, 0.2, 0.5, 0.7, 0.9\}$.

The β in exponential moving average for accumulating gradients is chosen to be 0.99 give more importance to historic gradients to mitigate the influence of latest potentially noisy gradients.

Note that batch-sizes, optimiser, learning rate and epochs mentioned in this table are also used for all comparative methods. Note that the shared layers optimiser doesn't utilise any momentum. This is in line with previous joint optimisation studies.

E PARAMETER SETTINGS IN BASELINES

Single Task Learning (STL): The SGD optimiser with momentum and learning rates mentioned in Table A2 are used for respective tasks.

Reptile: A slightly modified version of REPTILE, described in (Thakur et al., 2021), is used here to learn the shared model. Algorithm A1 illustrates this method. The learning rate described in Table A2 are used as inner learning rates, while the external learning rates, α, are chosen from this search space: $\{0.1, 0.2, 0.5, 0.75, 1\}$.

For patient care and ECG, $\alpha = 0.1$ is used. For discharge notes and image tasks, $\alpha = 0.2$, provided best performance on validation examples.

Algorithm A1 Reptile-based framework for training shared as well as task-specific parameters.

Input: $\mathcal{D}_t$: Dataset for task t, θ: shared parameters, ϕ_t: task-specific parameters for task t, α: outer learning rate, β: inner learning rate
for $t \leftarrow 1 : T$ **do** ▷ T: Number of Tasks
  $\mathbf{W}_t = \theta$
  $\mathcal{B} \leftarrow$SAMPLE-BATCHES$(\mathcal{D}_t)$ ▷ Batches
  for all $(\mathbf{b}, \mathbf{l}) \in \mathcal{B}$ **do**
    $\ell = \mathcal{L}(f_{\mathbf{W}_t, \phi_t}(\mathbf{b}), \mathbf{l})$
    $\mathbf{W}_t = \mathbf{W}_t - \beta \nabla_{\mathbf{W}_t} \ell$
    $\phi_t = \phi_t - \beta \nabla_{\phi_t} \ell$
  end for
end for
$\mathbf{G} = \frac{1}{T} \sum_{t=1}^{T} (\theta - \mathbf{W}_t)$
$\theta = \theta - \alpha \mathbf{G}$

iMAML: The batch-sizes used in other methods are also used here. 50 gradient steps with a regularisation term of 0.75 were used across all experiments. Whereas 10 conjugate gradient steps (Rajeswaran et al., 2019) were used to compute the meta-gradient in all experiment. The learning rates and optimiser mentioned in Table A2 are used for task-level inner updates and the outer learning rate of 0.01 was used for all task groups. The search space for outer learning rate was restricted to be $\{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 0.75, 1\}$.

KML: LSTM with 128 units was used as task encoder for patient care tasks. Whereas, the CRNN architecture is used as the task encoder for ECG tasks. For image classification tasks, the embedding generated by ResNet-50 before classification head is considered as task embedding. For NLP tasks, TinyBERT is used as task encoder. For all task groupings, the modulation network is an MLP with two fully connected layers, first followed by ReLU and next one by sigmoid. The output nodes are decided by the kernel size across different models. The rest of the parameters used in other baselines are also used here.

Linear model connectivity (LMC): The learning rates and batch sizes used in single task settings for all tasks are also used here. We use $\alpha = \{0.2, 0.4, 0.6, 0.8\}$ across all experiments. In each task grouping, the multiple tasks are learned in an incremental manner as described in Figure 15 of the supplementary document of Mirzadeh et al. (2021).

GradNorm: The only hyperparameter in this algorithm is the asymmetry or scaling coefficient α. We set the search space for α to be $\{0.1, 0.2, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3\}$. For patient care, we found 0.75 to be optimal. Similarly, $0.25, 1.5, 1.25$ were used for discharge notes, ECG and image tasks, respectively. The rest of the parameters mentioned in Reptile are also used here.

Mixture-of-Experts: Across all task groupings, we employed 8 experts, and top-4 experts were considered were selected for computing features. The rest of the parameters mentioned in Reptile are also used here.

F ADDITIONAL RESULTS: MNIST, Fashion MNIST & Kuzushiji-MNIST

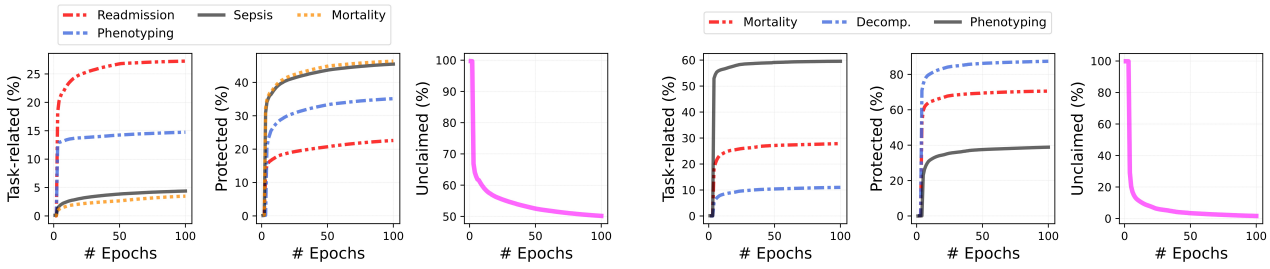
The performance of the proposed APO algorithm is also compared against the comparative methods on gray-scale image datasets. These datasets include MNIST, Fashion-MNIST, KUZUSHIJI-MNIST[1] datasets. These datasets are negatively associated with each other, making shared training difficult.

Table A3 documents the results of this experiment. The analysis of this highlights that the proposed APO and APO with mixture of experts outperform all baselines effectively.

[1] https://github.com/rois-codh/kmnist

Table A3: Evaluation of the proposed APO framework alongside existing approaches on the MNIST, Fashion MNIST, and Kuzushiji MNIST datasets.

Methods ↓	MNIST	Fashion MNIST	Kuzushiji-MNIST	Average
STL	98.83 ± 0.10	$\mathbf{90.93} \pm 0.15$	$\mathbf{93.20} \pm 0.10$	94.32 ± 0.12
REPTILE	98.53 ± 0.13	89.65 ± 0.06	90.07 ± 0.16	92.75 ± 0.12
iMAML	98.75 ± 0.07	89.93 ± 0.11	90.45 ± 0.18	93.04 ± 0.12
LMC	98.90 ± 0.05	89.46 ± 0.14	90.71 ± 0.14	93.02 ± 0.11
MoE	98.32 ± 0.13	88.93 ± 0.17	91.31 ± 0.10	92.85 ± 0.13
GradNorm	98.54 ± 0.06	89.65 ± 0.12	90.29 ± 0.15	92.83 ± 0.11
KML	98.72 ± 0.1	89.8 ± 0.09	90.12 ± 0.13	92.88 ± 0.11
PCGrad	98.61 ± 0.05	88.27 ± 0.17	90.10 ± 0.16	92.33 ± 0.13
Proposed	$\mathbf{99.06} \pm 0.09$	90.87 ± 0.13	93.15 ± 0.14	$\mathbf{94.36} \pm 0.12$
Proposed+MoE	98.86 ± 0.11	90.16 ± 0.16	91.56 ± 0.18	93.53 ± 0.15



(a) (b)

Figure A1: Percentage of parameters updated in the shared layers by each (a) discharge notes and (b) patient care task during training of models using the proposed algorithm

G DYNAMIC PARAMETER INTERPLAY IN APO

In the proposed APO framework, the lazy constraint associates distinct subsets of parameters within the shared layers to each task. Parameters that have been associated with other tasks are considered protected when updating the shared layers for the current task. Conversely, parameters that have not been updated by any task are considered unclaimed. At the beginning of training, all parameters are unclaimed. As training progresses, the number of parameters associated with each task grows.

This behaviour results in data-dependent regularisation, as the parameter conditioning for information sharing adjusts based on the data. As discussed in the main text, the increment in task-specific parameters during training is equivalent to increasing the effective degrees of freedom (DoF) of the model (Thakur et al., 2022). Initially, the DoF is lower, inducing stronger regularisation. As training progresses, each task acquires the effective DoF required for its convergence. Hence, APO induces a dynamic regularisation mechanism where the extent of regularisation is somewhat dependent on the task.

Figure A1 illustrates the evolution of task-specific and protective parameters in MIMIC-III and discharge notes tasks.

H RELATION TO CONTINUAL LEARNING

Continual Learning (CL) traditionally follows a *sequential* training paradigm, where tasks are learned in isolation and revisiting past tasks is rare. Knowledge transfer in CL primarily occurs through *forward transfer*, where earlier tasks influence later ones. However, *backward transfer*where later tasks improve earlier onesis generally limited (Benavides-Prado and Riddle, 2022), restricting continual adaptation across tasks.

In contrast, the proposed *Adaptive Parameter Optimisation* (APO) framework adopts a *cyclic* training paradigm that fosters continuous inter-task interaction. Within an epoch, tasks are processed in the order

$$\text{Task A} \rightarrow \text{Task B} \rightarrow \text{Task C},$$

Table A4: Performance comparison against the continual learning baselines on MIMIC-III patient care tasks.

Method	Phenotyping	Mortality	Decompensation	Average
EWC	0.671 (0.023)	0.818 (0.018)	0.821 (0.015)	0.770 (0.019)
SKILL	0.748 (0.012)	0.836 (0.017)	0.797 (0.016)	0.794 (0.015)
SUPSUP	0.716 (0.011)	0.821 (0.013)	0.769 (0.017)	0.773 (0.014)
PROPOSED	**0.792** (0.013)	**0.852** (0.016)	**0.832** (0.021)	**0.825** (0.017)

Table A5: Performance comparison against the continual learning baselines on discharge notes tasks.

Method	Readmission	Phenotyping	Sepsis	Mortality	Average
EWC	0.732 (0.018)	0.539 (0.014)	0.839 (0.022)	0.756 (0.017)	0.717 (0.017)
SKILL	0.862 (0.015)	0.586 (0.013)	0.921 (0.018)	0.753 (0.013)	0.781 (0.015)
SUPSUP	0.819 (0.017)	0.553 (0.016)	0.843 (0.013)	0.728 (0.011)	0.736 (0.014)
PROPOSED	**0.897** (0.019)	**0.608** (0.010)	**0.941** (0.012)	**0.758** (0.015)	**0.801** (0.014)

where *each task is conditioned on the preceding ones*. However, unlike CL, training does not progress strictly forward; in subsequent epochs, *Task A is revisited with knowledge updates from Task B and Task C of the previous cycle*. This iterative conditioning enables *bidirectional knowledge transfer*, where earlier tasks benefit from later ones, addressing a key limitation of CLs rigid sequential structure.

H.1 Comparison with CL Baselines

We benchmarked APO against three widely used CL methods: *Elastic Weight Consolidation* (EWC), *Structured Knowledge Injection for Lifelong Learning* (SKILL), and *Supermasks in Superposition* (SUPSUP) on patient-care and discharge notes task groups.

Notably, the high similarity of data across tasks in these domains posed challenges for SKILL and SUPSUPs task-mapping mechanisms. To mitigate this, we assumed the availability of task IDs during evaluation. Additionally, SKILL requires a pre-trained backbone, which is not inherently available for patient-care tasks. To address this, we pre-trained the backbone using self-supervised learning, segmenting time-series data and training the model to predict if two segments were consecutive. For discharge notes, we utilized a pre-trained TinyBERT model.

Tables A4 and A5 summarize the results, highlighting the following key observations:

- Elastic Weight Consolidation (EWC) exhibits significantly lower performance due to its inability to retain information from earlier tasks. This reinforces the limitations of purely sequential CL methods, particularly in settings where backward transfer is essential.
- While SKILL and SUPSUP prevent catastrophic forgetting using pre-trained feature extraction backbones (SKILL) and parameter isolation (SUPSUP), their knowledge transfer mechanisms remain largely *static*, as they do not actively optimize for inter-task interactions.
- Unlike traditional multi-task learning, APO prevents task interference while promoting active knowledge exchange by dynamically optimizing both shared and task-specific parameters in response to task-specific gradients.
- A major distinction of APO is its cyclic training paradigm. Unlike SKILL and SUPSUP, which adhere to a sequential learning approach, APO continuously revisits earlier tasks with updated knowledge, fostering effective backward transfer. This iterative refinement likely contributes to its superior performance.