The Dynamics of Task Interactions in Deep Linear Multi-Task Networks

Anonymous authors
Paper under double-blind review

Abstract

Despite significant empirical progress in multi-task learning (MTL), a theoretical understanding of task interactions and their learning dynamics remains limited. We address this gap by deriving analytical solutions for the gradient-flow dynamics of linear multi-task networks, characterising how shared and task-specific components evolve during training and at convergence. We show that task alignment and magnitude imbalance in the data determine how task-specific functions, losses, and neural representations evolve relative to one another, and that standard measures of negative task interference reflect this structure. Our analysis provides a theoretical explanation for the dynamic nature of task interactions and motivates dynamic, data-dependent loss weighting in linear MTL. We validate these predictions on the Multi-MNIST dataset and nonlinear networks. These results establish a theoretical foundation for understanding task interactions and pave the way toward principled algorithms for task weighting and grouping.

1 Introduction

Inspired by biological intelligence, Multi-Task Learning (MTL) is a paradigm where a model learns several tasks simultaneously to improve generalization through shared representations. Since its formalization (Caruana, 1997), MTL has achieved significant success in Deep Learning across fields, including Computer Vision (Liu et al., 2019; Leang et al., 2020; Zhang et al., 2023), Reinforcement Learning (He et al., 2024a; Sodhani et al., 2021), and Natural Language Processing (Zhao et al., 2019; 2020; Chen et al., 2024). While learning multiple tasks through a shared representation can improve generalization and data efficiency, negative interference between tasks can lead to degraded performance compared to Single-Task Learning (STL) (Zhang & Yang, 2021). As a result, there has been growing interest in *Task Grouping*, i.e., identifying which tasks should be trained together (Sherif et al., 2024; Fifty et al., 2021; Song et al., 2022), and *Dynamic Task Weighting Algorithms* that adapt each task's influence during training (Guo et al., 2025; He et al., 2024b; Liu et al., 2021).

Despite progress on the empirical front, our theoretical understanding of task interactions in MTL remains limited. As a consequence, existing empirical approaches for mitigating negative interference often rely on computationally expensive methods based on gradient-based heuristics, without a clear consensus on what task interactions truly are or when they emerge (Xin et al.; Elich et al., 2023; Lin et al., 2022).

To bridge this gap, we examine how task (mis)alignment, defined by the relationship between the singular vectors of task-specific input-output correlation matrices (Lampinen & Ganguli, 2018), shapes the dynamics of deep linear multi-task networks, focusing on learning speed, task importance, and representation learning. We derive exact solutions to the gradient flow learning dynamics of deep linear MTL networks as a function of task alignment, providing one of the few analytical models of MTL. This work extends the established single-task literature for deep linear networks (Saxe et al., 2014; 2019; Dominé et al., 2025) to the MTL setting, where the resulting analytical solutions are used to study the impact of training different tasks simultaneously on the learning dynamics.

Our analysis shows that the degree of alignment, together with differences in the tasks' magnitudes, determines whether tasks are learned sequentially or simultaneously and how relative task importance evolves

through learning. Leveraging the convergence properties of the analytical solutions, we compare the shared and task-specific representations at convergence with those obtained from training tasks independently and explain how alignment and magnitude differences shape the corresponding disparities. These theoretical predictions are validated using deep linear networks trained on both synthetic and more realistic datasets. We further show empirically that the insights derived from two-layer networks extend to deeper and ReLU networks. Taken together, these results provide a principled framework for understanding task interactions in MTL, laying the foundation for theoretically grounded dynamic task-weighting algorithms and task-grouping strategies.

Contributions. We make the following contributions:

- We analytically characterise the gradient-flow dynamics of shared and task-specific components in linear multi-task networks under a broad class of initializations (Section 2 and 3).
- Using our analytical solutions, we show that task alignment and magnitude imbalance control the timing and severity of inter-task interactions and the relative learning speed in linear MTL, providing a theory-driven motivation for dynamic loss weighting (Section 4).
- We develop predictive criteria based on task alignment and relative magnitudes to characterise transfer and to compare single-task and multi-task representations at convergence (Section 5).
- Through experiments, we show that task alignment remains predictive of task interactions beyond the linear setting, including in non-linear networks trained on the Multi-MNIST dataset (Section 6).

1.1 Related Work

A more expansive Related Work section is included in Appendix A.

Theory of multi-task learning. Early work recognised task similarity as an important factor underlying the benefits of MTL (Baxter, 2000; Ben-David & Schuller, 2003; Xue et al., 2007), but defined similarity through single-task models, which is suitable for classical methods yet difficult to extend to neural networks. Fewer works study the theoretical properties of MTL with neural networks, and those that do typically characterize MTL through sample-efficiency bounds (Maurer et al., 2016) or inductive bias (Lippl & Lindsey, 2024; Collins et al., 2024). Other work (Wu et al., 2020; Lampinen & Ganguli, 2018) uses task data alignment to identify *when* transfer can occur, but does not address how tasks interact or how these interactions evolve during training. In contrast, MTL optimization methods treat task interactions as dynamic (Ban & Ji, 2024; Liu et al., 2021) without a clear consensus on what task interactions are or when they emerge (Guo et al., 2025; Elich et al., 2023). To address this gap, we derive solutions to the gradient-flow dynamics of linear MTL networks, providing a theoretical characterisation of how task data alignment and magnitude differences influence the evolution of relative task importance and interactions during training.

Gradient flow. The continuous-time limit of gradient descent, known as gradient flow, has been extensively studied as an approximation for understanding the training dynamics of neural networks in STL. In the case of deep linear networks, exact analytical solutions under tabula rasa (random) initialization and whitened inputs were first introduced by Saxe et al. (2014). These solutions revealed key features of learning dynamics, including layer-wise alignment of weights and the sequential learning of input-output modes via singular values. Later works extended gradient flow analyses to more general settings, such as varying network architectures (Saxe et al., 2022; Pinson et al., 2023), alternative initialization schemes (Tarmoun et al., 2021), and structured input data (Gidel et al., 2019). Recently, a line of work has adopted a Riccati-based formulation originally proposed by Fukumizu (1998), where the gradient flow equations for two-layer linear networks are recasted as solvable matrix differential equations (Yan et al., 1994). This approach has been extended to accommodate richer initialization structures (Tarmoun et al., 2021; Dominé et al., 2025), and task transfer scenarios (Braun et al., 2022), while a corresponding treatment of MTL has remained absent.

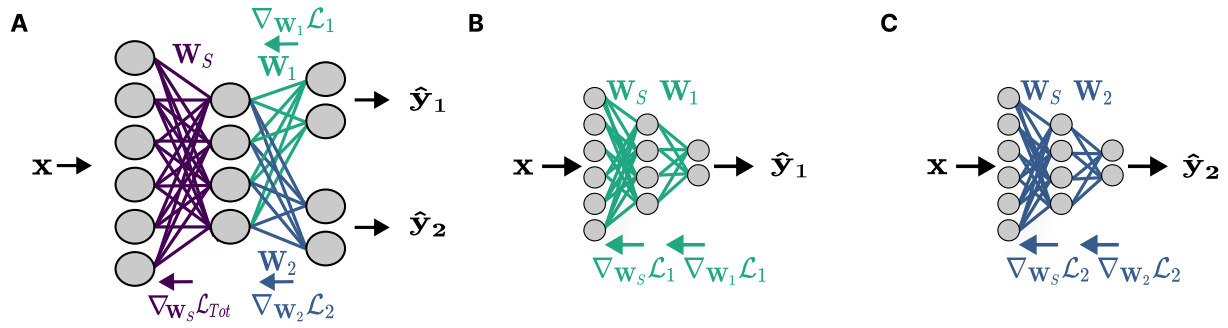

Figure 1: Difference between (**A**) Multi-Task Learning (MTL, our setup) and (**B**-**C**) Single-Task Learning (STL). MTL separates shared $\mathbf{W}_S$ (purple) and task-specific weights $\mathbf{W}_1$ (green), $\mathbf{W}_2$ (blue) with independent gradients $\nabla_{\mathbf{W}_1}\mathcal{L}_1$ and $\nabla_{\mathbf{W}_2}\mathcal{L}_2$. In STL, no explicit distinction is made between task(s).

2 Preliminary: MTL and Task Alignment

This work considers a Multi-Task Learning (MTL) (Ruder, 2017) setup in which the objective is to simultaneously learn N_T tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_{N_T}\}$. All tasks share a common input vector $\mathbf{x} \in \mathbb{R}^{N_1}$, where N_1 is the input dimensionality, but each task $\mathcal{T}_i$ is associated with task-specific output vectors $\mathbf{y}_i \in \mathbb{R}^{N_{io}}$, where N_{io} denotes the output dimensionality of task $\mathcal{T}_i$. To this end, a standard hard-parameter sharing architecture is adopted (Zhang & Yang, 2021). In this setup, the tasks are trained jointly using a shared map $\mathbf{W}_S = \mathbf{W}_S^{(L_S)} \cdots \mathbf{W}_S^{(1)}$, followed by $\mathcal{T}_i$-specific maps $\mathbf{W}_i = \mathbf{W}_i^{(L_i)} \cdots \mathbf{W}_i^{(1)}$, so that task i implements $\mathbf{W}_i\mathbf{W}_S$. The network is trained by minimizing a composite loss $\mathcal{L}_{\text{Tot}}$, a weighted sum of task-specific losses $\mathcal{L}_i$,

$$\mathcal{L}_{\text{Tot}}(t) = \sum_{i=1}^{N_T} \omega_i(t)\, \mathcal{L}_i(\mathbf{W}_i, \mathbf{W}_S, t), \tag{1}$$

where $\omega_i(t)$ is the (possibly time-dependent) weighting coefficient for task $\mathcal{T}_i$, controlling its relative contribution at training time t. We assume uniform static weighting $\omega_i = 1$ throughout, and discuss the effect of non-uniform weighting ($\omega_i \neq 1$) in Appendix D.

In this work, the neural network is trained with full-batch gradient descent with a mean-squared-error (MSE) task-specific loss function $\mathcal{L}_i^{\text{MSE}}$ for each task,

$$\mathcal{L}_i^{\text{MSE}} = \frac{1}{2N} \sum_{\mu=1}^{N} \left\| \mathbf{y}_i^\mu - \mathbf{W}_i\mathbf{W}_S\mathbf{x}^\mu \right\|^2, \tag{2}$$

where N is the number of input samples and $\mathbf{x}^\mu$, $\mathbf{y}_i^\mu$ denote the μ-th input and corresponding target output for task $\mathcal{T}_i$. While the shared weights $\mathbf{W}_S$ are updated based on the gradient computed on the total loss $\mathcal{L}_{\text{Tot}}(t)$ (equation 1), the task-specific weights are updated through task-specific gradients computed on the task-specific losses $\mathcal{L}_i$ and learning rate η according to the update rule,

$$\mathbf{W}_S^\ell(t+1) \leftarrow \mathbf{W}_S^\ell(t) - \eta \nabla_{\mathbf{W}_S^\ell(t)} \mathcal{L}_{\text{Tot}}(t) \tag{3}$$

$$\mathbf{W}_i^\ell(t+1) \leftarrow \mathbf{W}_i^\ell(t) - \eta \nabla_{\mathbf{W}_i^\ell(t)} \mathcal{L}_i(t). \tag{4}$$

As a result, task signals are explicitly disentangled in the task-specific layers, in contrast to single-task learning (STL), where task signals are not explicitly separated (Figure 1). In Appendix B.2, we discuss how our framework differs from existing STL theory and how it applies to STL settings.

2.1 MTL Gradient Flow Equations in Two-Layer Linear Neural Networks

As in prior analyses of STL dynamics (Saxe et al., 2014; 2019; Braun et al., 2022), we start with a two-layer linear neural network to obtain tractable training dynamics. In the corresponding MTL setup, tasks are

trained jointly using one shared hidden weight matrix $\mathbf{W}_S \in \mathbb{R}^{N_S \times N_1}$, where N_S denotes the dimension of the shared layer, followed by task-specific weight matrices $\mathbf{W}_i \in \mathbb{R}^{N_{io} \times N_S}$ for each task $\mathcal{T}_i$ (Figure 1), yielding the task-specific predicted outputs,

$$\hat{\mathbf{y}}_i = \mathbf{W}_i \mathbf{W}_S \mathbf{x} \qquad \forall i \in \{1, \dots, N_T\}. \tag{5}$$

In order to study the dynamics of the task-specific functions under gradient descent, we consider the continuous-time limit of equations (3) and (4), while explicitly distinguishing between different tasks,

$$\tau \frac{d\mathbf{W}_S}{dt} = \sum_{i=1}^{N_T} \mathbf{W}_i^\top (\mathbf{\Sigma}_{io} - \mathbf{W}_i \mathbf{W}_S \mathbf{\Sigma}_{11}) \tag{6}$$

$$\tau \frac{d\mathbf{W}_i}{dt} = (\mathbf{\Sigma}_{io} - \mathbf{W}_i \mathbf{W}_S \mathbf{\Sigma}_{11}) \, \mathbf{W}_S^\top, \tag{7}$$

$\forall i \in \{1, \dots, N_T\}$, where $\tau \equiv \frac{1}{\eta}$ and the input-output correlation matrices are defined as,

$$\mathbf{\Sigma}_{11} \equiv \frac{1}{N} \sum_{\mu=1}^{N} \mathbf{x}^\mu (\mathbf{x}^\mu)^\top \quad \text{and} \quad \mathbf{\Sigma}_{io} \equiv \frac{1}{N} \sum_{\mu=1}^{N} \mathbf{y}_i^\mu (\mathbf{x}^\mu)^\top,$$

where N represents the batch size. In Section 3, we rewrite equations (6)-(7) as Riccati equations (Fukumizu, 1998), enabling analytical solutions while explicitly preserving task distinctions. We use these solutions to describe how different tasks impact each other's learning speed (Section 4) and network representations (Section 5). To this end, we introduce task alignment and magnitude imbalance in subsection 2.2.

2.2 Task Alignment, Magnitude Imbalance and Shared MTL Spectrum

Central to the formal definition of task alignment are the task-specific input-output correlation matrices $\mathbf{\Sigma}_{io}$ and their singular value decomposition (SVD),

$$\mathbf{\Sigma}_{io} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top, \tag{8}$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ denote the left- and right $\mathcal{T}_i$-specific singular vectors and $\mathbf{S}_i$ is the diagonal matrix containing the singular values s_i^β with $s_i^1 \geq s_i^2 \geq \cdots \geq s_i^{N_{io}}$ which capture the strength of the $\mathcal{T}_i$-singular mode β. The amount of task alignment, as previously defined by Lampinen & Ganguli (2018), is quantified through the overlap matrix $\mathbf{O}_{12}$ defined below. We extend their definition to explicitly incorporate conflicting tasks.

Definition 2.1. *Consider two tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ defined by $\mathbf{\Sigma}_{1o}$ and $\mathbf{\Sigma}_{2o}$, with dimensions satisfying $N_{o_1} + N_{o_2} < N_1$. Let $\mathbf{V}_i \in \mathbb{R}^{N_1 \times N_{o_i}}$ collect the right singular vectors of $\mathbf{\Sigma}_{io}$ associated with its N_{o_i} nonzero singular values (equation 8). The overlap matrix $\mathbf{O}_{12}$ is defined as*

$$\mathbf{O}_{12} = \mathbf{V}_1^\top \mathbf{V}_2 \in \mathbb{R}^{N_{o_1} \times N_{o_2}}, \tag{9}$$

with singular values $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0$, where $r = \min(N_{o_1}, N_{o_2})$. Two tasks are said to be: (i) Fully Aligned if $\alpha_i = 1$, $\forall i$; (ii) Fully Orthogonal if $\alpha_i = 0$, $\forall i$; and (iii) Fully Conflicting when $0 < \alpha_i < 1$, $\forall i$.

In Section 3, we show that the MTL gradient flow equations (6)-(7) can be reformulated and solved in terms of the combined task space ξ and its singular value decomposition,

$$\xi = \begin{pmatrix} \mathbf{\Sigma}_{1o} & \mathbf{\Sigma}_{2o} \end{pmatrix}^\top = \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^\top. \tag{10}$$

The diagonal entries of $\mathbf{S}_\xi$ are given by the square roots of the eigenvalues of

$$\mathbf{M} = \xi^T \xi = \mathbf{V}_1 \mathbf{S}_1^2 \mathbf{V}_1^\top + \mathbf{V}_2 \mathbf{S}_2^2 \mathbf{V}_2^\top, \tag{11}$$

indicating that the spectrum $\mathbf{S}_\xi$ reflects the degree of (mis)-alignment between the singular modes of each task (Figure 2). For fully aligned tasks $\mathbf{V}_1 = \mathbf{V}_2 = \mathbf{V}$, such that $\mathbf{M} = \mathbf{V}(\mathbf{S}_1^2 + \mathbf{S}_2^2)\mathbf{V}^\top$ and Rank($\mathbf{S}_\xi$)=Rank($\mathbf{S}_i$).

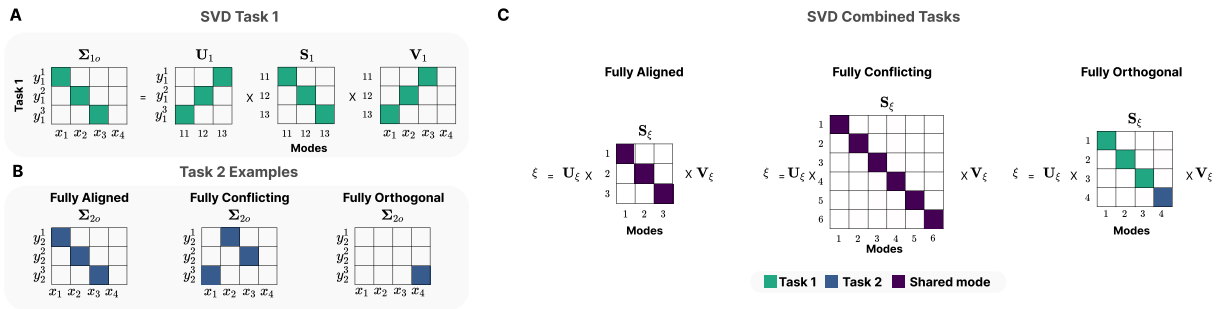

Figure 2: Example Singular Value Decompositions after combining two tasks into a combined task space ξ. **A**. SVD of the input-output correlation matrix of Task 1. **B**. Example input-output correlation matrices for a fully aligned, conflicting, and orthogonal Task 2 (relative to Task 1). **C**. SVDs resulting from combining Task 1 with each distinct Task 2. Aligned tasks reinforce (shared modes strengthen), conflicting tasks interfere (modes mix across shared modes), and orthogonal tasks remain independent (each mode maps to one task).

In contrast, when the tasks are fully orthogonal $\text{span}(\mathbf{V}_1) \perp \text{span}(\mathbf{V}_2)$, such that the spectrum of eigenvalues corresponds to the ordered union of singular values from both tasks, i.e., $\text{Rank}(\mathbf{S}_\xi)$=$\text{Rank}(\mathbf{S}_1 \oplus \mathbf{S}_2)$. For conflicting tasks, the spectrum is neither a simple union nor a fully shared spectrum, but a mode-wise mixture in which contributions from both tasks combine according to their degree of alignment α. For two one-dimensional tasks with equal strengths $s_1^1 = s_2^1 = s$, for instance, the two task strengths combine into a stronger cooperative and a weaker conflicting mode ($s\sqrt{1+\alpha}$ and $s\sqrt{1-\alpha}$). In Section 6 and Appendix D.2, we consider tasks that do not fit neatly into the categories of Definition 2.1 and show that our analysis still applies.

Magnitude Imbalances. In addition to task alignment, we analyse the effect of magnitude disparities between tasks. We define the magnitude of task i as the Frobenius norm of its input-output correlation matrix, $M_i = \|\mathbf{\Sigma}_{io}\|_F = \sqrt{\sum_\beta (s_i^\beta)^2}$, and quantify the imbalance between two tasks $(\mathcal{T}_i, \mathcal{T}_j)$ by the ratio

$$\Delta M_{ij} = \frac{\min(M_i, M_j)}{\max(M_i, M_j)} \in (0, 1], \tag{12}$$

so that $\Delta M_{ij} = 1$ for balanced tasks and $\Delta M_{ij} \to 0$ as one task dominates.

3 Exact Solutions and Inter-Task Dynamics

To obtain exact solutions of the gradient-flow dynamics, we extend the Riccati-based formulation of Fukumizu (1998) to the multi-task setting by rewriting the gradient-flow equations (6)–(7) as a matrix Riccati equation. This formulation has previously been used to characterise single-task dynamics (Tarmoun et al., 2021; Braun et al., 2022). We derive the solution for the two-task case ($N_T = 2$) and discuss the general case $N_T > 2$ in Appendix C.

This formulation allows for an explicit computation of the evolution of the network's shared and task-specific representations. These quantities will be used in Sections 4 and 5 to analyse how task alignment and magnitude imbalance (Section 2.2) affect both learning speed and the structure of the learned representations in linear MTL networks.

In order to write the MTL dynamics as a Riccati matrix equation, we define, $\mathbf{Q}_{MTL} = \begin{pmatrix}\mathbf{W}_S^\top & \mathbf{W}_1 & \mathbf{W}_2\end{pmatrix}^\top$ and consider,

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t) = \begin{pmatrix} \mathbf{W}_S^\top\mathbf{W}_S(t) & \mathbf{W}_S^\top\mathbf{W}_1^\top(t) & \mathbf{W}_S^\top\mathbf{W}_2^\top(t) \\ \mathbf{W}_1\mathbf{W}_S(t) & \mathbf{W}_1\mathbf{W}_1^\top(t) & \mathbf{W}_1\mathbf{W}_2^\top(t) \\ \mathbf{W}_2\mathbf{W}_S(t) & \mathbf{W}_2\mathbf{W}_1^\top(t) & \mathbf{W}_2\mathbf{W}_2^\top(t) \end{pmatrix} \qquad \mathbf{R}_{MTL} = \begin{pmatrix} 0 & \mathbf{\Sigma}_{1o}^\top & \mathbf{\Sigma}_{2o}^\top \\ \mathbf{\Sigma}_{1o} & 0 & 0 \\ \mathbf{\Sigma}_{2o} & 0 & 0 \end{pmatrix},$$

such that, $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$ contains all necessary information to compute task-specific network functions $\mathbf{W}_i\mathbf{W}_S(t)$. Lastly, define $\mathbf{\Gamma} = \lambda\mathbf{I}$, such that $\mathbf{\Gamma}$ is a constant matrix describing the difference in covariance between the shared and task-specific weights.

The derivation of the Riccati equation and its analytical solution rely on the standard assumptions made in STL (Braun et al., 2022; Dominé et al., 2025; Fukumizu, 1998):

- **Assumption A1.** The inputs are *whitened*, i.e., $\mathbf{\Sigma}_{11} = \mathbf{I}$.
- **Assumption A2.** The weights are *zero-balanced* at initialization: $\mathbf{W}_S(0)\mathbf{W}_S(0)^\top = \mathbf{W}_1(0)^\top\mathbf{W}_1(0) + \mathbf{W}_2(0)^\top\mathbf{W}_2(0)$. **A2bis.** The weights are $\mathbf{\Gamma}$-*balanced* at initialization: $\mathbf{W}_S(0)\mathbf{W}_S(0)^\top - \mathbf{W}_1(0)^\top\mathbf{W}_1(0) - \mathbf{W}_2(0)^\top\mathbf{W}_2(0) = \mathbf{\Gamma}$.
- **Assumption A3.** *No bottleneck:* $\forall$ i $\in$ {1,2}: Rank($\mathbf{\Sigma}_{io}$)=min(N_1, N_{io}), ensuring the shared space can hold all the task-relevant input-output modes for both tasks $N_S \geq \min(N_1, N_{1o} + N_{2o})$.

Under these assumptions, the MTL dynamics governed by $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top$ and $\mathbf{R}_{MTL}$ can be described by a system of Matrix Riccati equations, as formalized in Lemma 3.1. A detailed discussion of these assumptions is provided in Appendix C.1 and Dominé et al. (2025); Jarvis et al. (2025).

Lemma 3.1. *Under assumptions A1 (whitened inputs) and A2bis (Γ-balanced weights), the continuous-time dynamics of the matrix $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$ satisfy the matrix Riccati Equation,*

$$\tau\frac{d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)}{dt} = \mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top + \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} - (\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2 - \mathbf{Q}_{MTL}\mathbf{\Gamma}\mathbf{Q}_{MTL}^\top.$$

The proof of Lemma 3.1 is given in Appendix C.2. In order to obtain an interpretable solution of the Riccati equation in Lemma 3.1, we define,

$$\xi = \begin{pmatrix}\mathbf{\Sigma}_{1o} & \mathbf{\Sigma}_{2o}\end{pmatrix}^\top \qquad \mathbf{W}_{MTL}^0 = \begin{pmatrix}\mathbf{W}_1(0)\mathbf{W}_S(0) & \mathbf{W}_2(0)\mathbf{W}_S(0)\end{pmatrix}^\top,$$

such that ξ and $\mathbf{W}_{MTL}^0 \in \mathbb{R}^{(N_{1o}+N_{2o})\times N_1}$, allow us to consider the compact singular value decomposition of the initial network and the input-output correlation matrix of the combined task space,

$$\xi = \mathbf{U}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^\top \qquad \mathbf{W}_{MTL}^0 = \mathbf{U}_0\mathbf{S}_0\mathbf{V}_0^\top. \tag{13}$$

When $N_{1o} + N_{2o} < N_1$, the factors $\mathbf{U}_\xi$ and $\mathbf{V}_\xi$ have orthonormal columns but are not square, and hence do not form complete orthonormal bases. Following Braun et al. (2022), we construct the complements $\mathbf{U}_\xi^\perp$ and $\mathbf{V}_\xi^\perp$ to complete the bases (Appendix C.3). We then introduce

$$\mathbf{A}_\xi = \mathbf{U}_0^\top\mathbf{U}_\xi + \mathbf{V}_0^\top\mathbf{V}_\xi, \qquad \mathbf{B}_\xi = \mathbf{U}_0^\top\mathbf{U}_\xi - \mathbf{V}_0^\top\mathbf{V}_\xi, \tag{14}$$

which allow us to formulate the solution for $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$ in Theorem 3.2.

Theorem 3.2. *Under assumptions A1 (whitened inputs), A2 (zero-balanced weights), A3 (no-bottleneck) and non-singular $\mathbf{A}_\xi$, the MTL Riccati equation (Lemma 3.1) has a unique solution $\forall t > 0$, given by,*

$$\begin{aligned}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t) = \ \mathbf{Z}_{MTL}\Bigg[& 4\,e^{-\mathbf{S}_\xi\frac{t}{\tau}}\,\mathbf{A}_\xi^{-1}\,\mathbf{S}_\xi^{-1}\left(\mathbf{A}_\xi^\top\right)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}} \ + \ \left(\mathbf{I} - e^{-2\mathbf{S}_\xi\frac{t}{\tau}}\right)\mathbf{S}_\xi^{-1} \\ & - e^{-\mathbf{S}_\xi\frac{t}{\tau}}\,\mathbf{A}_\xi^{-1}\,\mathbf{B}\left(e^{-2\mathbf{S}_\xi\frac{t}{\tau}} - \mathbf{I}\right)\mathbf{S}_\xi^{-1}\mathbf{B}^\top\left(\mathbf{A}_\xi^\top\right)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}} \\ & + 4\,\frac{t}{\tau}\,e^{-\mathbf{S}_\xi\frac{t}{\tau}}\,\mathbf{A}_\xi^{-1}\left(\mathbf{V}_\xi^\top\,\mathbf{V}_\xi^\perp(\mathbf{V}_\xi^\perp)^\top\,\mathbf{V}_\xi \ + \ \mathbf{U}_\xi^\top\,\mathbf{U}_\xi^\perp(\mathbf{U}_\xi^\perp)^\top\,\mathbf{U}_\xi\right)\left(\mathbf{A}_\xi^\top\right)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}}\Bigg]^{-1}\mathbf{Z}_{MTL}^\top,\end{aligned}$$

with

$$\mathbf{Z}_{MTL} = \begin{pmatrix}\mathbf{V}_\xi\left(\mathbf{I} - e^{-\mathbf{S}_\xi\frac{t}{\tau}}\mathbf{B}_\xi^\top(\mathbf{A}_\xi^\top)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}}\right) + 2\,\mathbf{V}_\xi^\perp(\mathbf{V}_\xi^\perp)^\top\mathbf{V}_\xi(\mathbf{A}_\xi^\top)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}} \\ \mathbf{U}_\xi\left(\mathcal{I} + e^{-\mathbf{S}_\xi\frac{t}{\tau}}\mathbf{B}_\xi^\top(\mathbf{A}_\xi^\top)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}}\right) + 2\,\mathbf{U}_\xi^\perp(\mathbf{U}_\xi^\perp)^\top\mathbf{U}_\xi(\mathbf{A}_\xi^\top)^{-1}e^{-\mathbf{S}_\xi\frac{t}{\tau}}\end{pmatrix}.$$

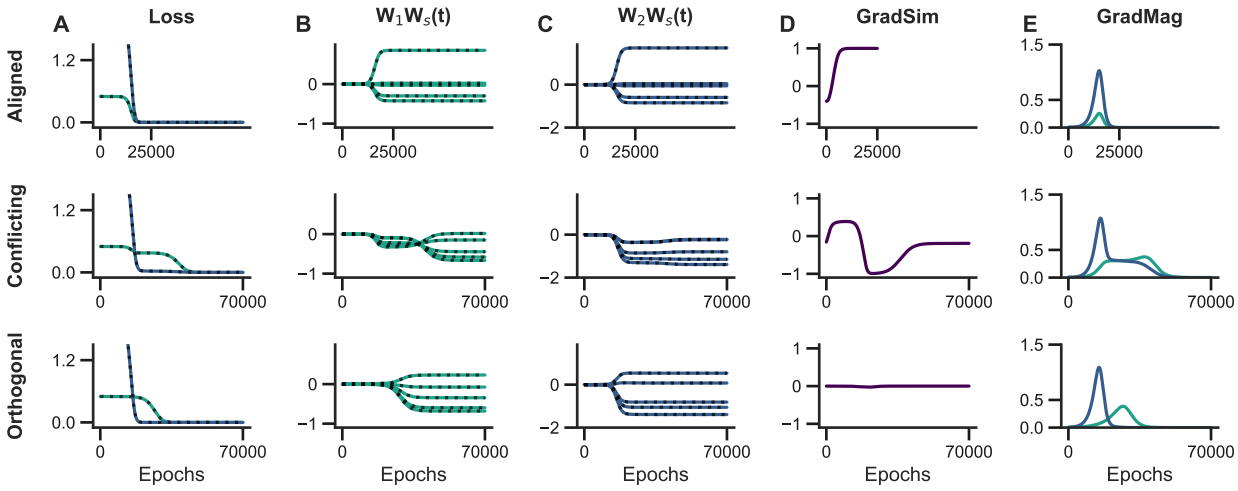

Figure 3: **Column A.** Analytical solutions (black dotted lines) match numerical simulations of task-specific losses (task 1: green; task 2: blue solid lines) and, **Columns B-C**, the components of the network functions $\mathbf{W}_i\mathbf{W}_S$ for a linear neural network ($N_1 = 5$, $N_S = 2$, $N_{o1} = N_{o2} = 1$). The network is trained on fully aligned (top row), conflicting (middle row), and orthogonal (bottom row) regression tasks constructed as described in Appendix D.1, with magnitude imbalance $\Delta M_{1,2} = 0.5$. **Columns D-E.** Corresponding cosine similarity between task-specific gradients (GradSim, Eq. 16) and gradient magnitudes (GradMag, Eq. 15) for both tasks over training until convergence. In Appendix D.1, we compare these dynamics with those observed in the analogous balanced-magnitude setting and in ReLU networks.

For completeness, the proof of the STL version of Theorem 3.2 is given in Braun et al. (2022), while the key modifications required to extend the argument to the multi-task setting are outlined in Appendix C.3. We outline the steps required to generalise our results to λ-balanced initialisations (relaxing the zero-balanced Assumption A2) in Appendix C.6, and to bottlenecked networks (relaxing Assumption A3) in Appendix C.7.

Theorem 3.2 shows that MTL dynamics depend on the MTL spectrum $\mathbf{S}_\xi$, which encodes task (mis)alignment (Section 2.2), indicating that task alignment plays an important role in training dynamics. We verify that the analytical solution matches numerical simulations of the training dynamics (Figure 3). Building on this, we analyse how task alignment (Definition 2.1) shapes learning in linear MTL networks (Section 4). Section 5 leverages Theorem 3.2 to directly compare multi-task and single-task representations at convergence, without additional training. We empirically validate these predictions on ReLU networks trained on Multi-MNIST in Section 6. Experimental details are in Appendix D, with code provided in the supplemental material and released online.[1]

4 Task Alignment, Magnitude Differences and Learning Dynamics

From single-task literature (Braun et al., 2022; Saxe et al., 2014), we know that a linear network's input-output map sequentially learns the task-specific singular modes. The speed at which each singular mode β is learned is dependent on the strength of the mode, which is reflected by the corresponding singular value s^{β}[2]. Theorem 3.2 shows that in MTL, these singular values are effectively captured by the diagonal matrix $\mathbf{S}_\xi$ of the shared MTL spectrum. As a result, the learning speed of the modes and the evolution of the task-specific network functions are determined by the alignment of the tasks (Section 2.2). In addition, prior MTL work indicates that unequal task scaling can degrade performance (Chen et al., 2018; Ban & Ji, 2024). We therefore examine whether magnitude disparities between tasks (i.e., $\Delta M_{ij} \neq 1$) bias the training

[1] A link to the GitHub repository will be shared here upon acceptance.

[2] Appendix A provides a detailed discussion of earlier results from the literature, with step-by-step derivations of these results given in Appendix B.

trajectory toward the higher-magnitude task, causing it to (temporarily) dominate learning, and how this effect interacts with task alignment.

To study the effects of task alignment and magnitude imbalance, we use the analytical solution of Theorem 3.2 to simulate the learning dynamics (Figure 3). We construct task pairs with controlled alignment and fixed singular values (Appendix D.1) and analyse the resulting dynamics under imbalanced task magnitudes ($\Delta M_{12} = 0.5$). We validate these predictions by comparing the analytical solutions with the network functions and task-specific losses obtained from training a linear MTL network on the same tasks. In Appendix D.2, we extend the analysis to higher-dimensional tasks with mixed spectra that do not fit neatly into the fully aligned, conflicting, and orthogonal categories presented here.

Perfect alignment enables simultaneous convergence. When tasks are fully aligned, each pair of task-specific singular values s_1^β, s_2^β combines in quadrature into a single shared mode for each β (Section 2.2). Each combined mode is therefore shared by both tasks and evolves on a single common timescale. As observed in Figure 3 (first row), this produces a perfectly simultaneous evolution of the task-specific losses (Figure 3**A**) and network functions $\mathbf{W}_i\mathbf{W}_S(t)$ (Figure 3**B**-**C**): both tasks begin learning and converge together. Because the shared mode sets a single timescale, this simultaneity is independent of the relative task magnitudes, holding for both imbalanced (Figure 3) and balanced (Figure 11) magnitudes.

Orthogonality implies task-by-task mode learning. Because the singular modes of orthogonal tasks do not couple, each mode in $\mathbf{S}_\xi$ is picked up by a single task's network function while the other remains unchanged (Figure 3, bottom row). Since the combined spectrum is the union of the two task spectra, modes resolve in order of their strength. If all singular modes of one task are stronger than those of the other, that task converges first, delaying the learning of the second. As seen in Figure 3**A**, this produces a clear gap between the convergence of the two task-specific losses and network functions (Figure 3**B**-**C**), appearing as two distinct learning stages since both tasks are rank one ($N_{o_1} = N_{o_2} = 1$). This gap is set by the difference in task magnitudes, and shrinks as that imbalance is reduced (Figure 11).

Conflicting tasks cooperate before they compete. For fully conflicting tasks (Figure 3, middle row), the two tasks' individual singular modes are partially shared: with overlap $0 < \alpha < 1$, the combined spectrum $\mathbf{S}_\xi$ splits into a stronger cooperative mode and a weaker conflicting mode (Section 2.2). Since the cooperative mode is stronger, it resolves first, so both losses initially decrease together (as for fully aligned tasks). They then plateau while the weaker conflicting mode resolves, before converging (Figure 3**A**). The length of this plateau is governed by the overlap α: a larger α widens the gap between the cooperative and conflicting mode strengths, prolonging the plateau before the conflicting mode resolves. Because the tasks are magnitude-imbalanced ($\Delta M_{12} = 0.5$), one task (Figure 3**B**) has more remaining structure to learn than the other (Figure 3**C**). We provide further experiments across different degrees of conflict in Appendix D.1.1.

Theoretical motivation for weighting tasks dynamically. The preceding analysis shows that inter-task interactions in linear MTL are inherently dynamic, driven by task misalignment and magnitude imbalance in the data, which provides a theoretical motivation for dynamic loss weighting. To connect this to existing practice, we examine two measures of negative interference commonly used in the MTL literature Ban & Ji (2024); Guo et al. (2025); Liu et al. (2021): imbalanced gradient magnitudes,

$$\mathbf{GradMag}_i = \|\nabla_{\mathbf{W}_S}\mathcal{L}_i\|, \tag{15}$$

and conflicting gradient directions, quantified by gradient cosine similarity,

$$\mathbf{GradSim} = \frac{\nabla_{\mathbf{W}_S}\mathcal{L}_1 \cdot \nabla_{\mathbf{W}_S}\mathcal{L}_2}{\|\nabla_{\mathbf{W}_S}\mathcal{L}_1\|\,\|\nabla_{\mathbf{W}_S}\mathcal{L}_2\|}. \tag{16}$$

As shown in Figure 3(**D**-**E**), both measures vary across alignment regimes in a way that directly reflects the combined spectrum. Fully aligned tasks (Figure 3, top row) exhibit consistently positive cosine similarity and a single shared rise in gradient magnitude, reflecting the single shared mode. Fully orthogonal tasks (Figure 3, bottom row) maintain near-zero cosine similarity during learning, with gradient magnitudes rising at separated times, one per independent mode. Conflicting tasks (Figure 3, middle row) display intermediate, time-varying behaviour: the cosine similarity transitions from positive to negative, and the gradient magnitudes show an initial shared rise before the two tasks separate, reflecting the split of the

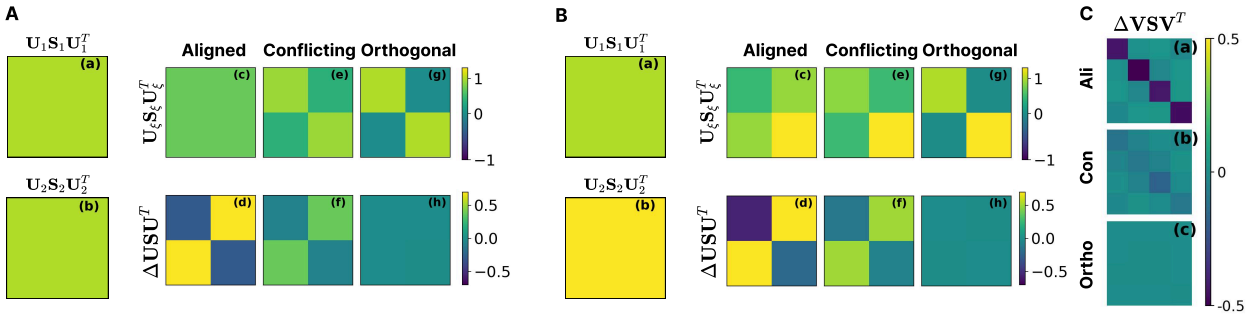

Figure 4: Comparison of RSMs for regression task pairs ($N_1 = 4, N_{o1} = N_{o2} = 1$) with different alignment constructed as described in Appendix D.1. **A**. Balanced tasks ($\Delta M_{12} = 1$), where single task RSMs $\mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top$ combine into $\mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^\top$. Their corresponding difference $\Delta\mathbf{USU}^\top$ shows weaker structural relationships for fully aligned and conflicting tasks, but no effect from fully orthogonal tasks as compared to the single task case. **B**. These effects are amplified by a magnitude difference $\Delta M_{12} = 0.25$. **C**. $\Delta\mathbf{VSV}^\top$. Aligned tasks interfere and create different structural relationships, conflicting tasks show intermediate effects, while for orthogonal tasks, this yields exactly the direct sum between the task-specific input RSMs.

combined spectrum into a cooperative and a conflicting mode. These standard measures thus recover the same regime structure predicted by our analysis, confirming that the strength and timing of interference are set by task alignment and magnitude imbalance.

Together, these results show that task alignment and magnitude imbalance jointly determine the relative learning speeds of the tasks and the degree to which inter-task interference evolves during training. In Section 6, we further investigate whether these observations extend to nonlinear networks. Results for intermediate degrees of alignment (Figure 13) and non-whitened inputs (Figure 15) are provided in Appendix D.1.

5 Shared- and Task-specific Representations At Convergence

We now examine the impact of task alignment on the MTL network representations at convergence. To this end, we use the following convergence theorem (Theorem 5.1).

Theorem 5.1. *Under the assumptions of Theorem 3.2,* $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$ *converges to a steady state given by,*

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t_{steady}) = \begin{pmatrix} \mathbf{V}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^T & \mathbf{V}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^T \\ \mathbf{U}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^T & \mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^T \end{pmatrix}.$$

This result is a direct consequence of Theorem 3.2. For completeness, the proof is included in Appendix C.4. Theorem 5.1 states that the Gram Matrices of the shared and task-specific weights ($\mathbf{W}_S^\top\mathbf{W}_S$ and $\mathbf{W}_i\mathbf{W}_i^\top$ respectively), converge to the Representation Similarity Matrices (RSMs) of the combined task data ξ, i.e.,

$$\lim_{t\to t_{steady}} \mathbf{W}_S^\top\mathbf{W}_S(t) = \mathbf{V}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^\top \tag{17}$$

$$\lim_{t\to t_{steady}} \mathbf{W}_i\mathbf{W}_i^\top(t) = \mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^\top \left[\sum_{j<i} N_{jo} : \sum_{j\leq i} N_{jo}\right] \tag{18}$$

Here, $\mathbf{V}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^T$ and $\mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^T$, are the RSMs of the combined input and output data, respectively. These matrices encode the similarities between different input and output dimensions and hence represent structural relationships in the input and output data. In Figure 4, we show that task alignment and magnitude imbalance affect the RSMs of the combined input and output data. As a consequence of Theorem 5.1, these factors also impact the way the network's weights reflect structural relationships from the tasks.

Theorem 5.1 also enables a direct theoretical comparison between the task-specific and shared representations learned in MTL and those learned when training each task independently without requiring model training.

Namely, from the STL convergence theorem in Braun et al. (2022), it follows that the hidden and output weights in a STL network initialized with zero-balanced weights converge to the task-i-specific RSMs of the input ($\mathbf{V}_i\mathbf{S}_i\mathbf{V}_i^\top$) and the output ($\mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top$),

$$\lim_{t\to t_{steady}} (\mathbf{W}_S^{STL})^\top \mathbf{W}_S^{STL}(t) = \mathbf{V}_i\mathbf{S}_i\mathbf{V}_i^\top \tag{19}$$

$$\lim_{t\to t_{steady}} \mathbf{W}_i^{STL}(\mathbf{W}_i^{STL})^\top(t) = \mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top. \tag{20}$$

As a consequence, equations (17)-(18) and (19)-(20) allow comparison of how linear MTL networks reflect structural relationships at convergence when trained jointly on multiple tasks versus separately on each task. In Figures 4**A** and 4**B**, we visualize the difference between the RSMs of the output data by defining $\mathbf{\Delta USU}^\top = (\mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^T)_{[:N_{1o},:N_{1o}]} - \mathbf{U}_1\mathbf{S}_1\mathbf{U}_1^\top$. To quantify how strong the combined structure deviates from the direct sum of the task specific RSMs of the input data, we define $\mathbf{\Delta VSV}^\top = \mathbf{V}_\xi\mathbf{S}_\xi\mathbf{V}_\xi^\top - \mathbf{V}_1\mathbf{S}_1\mathbf{V}_1^\top \oplus \mathbf{V}_2\mathbf{S}_2\mathbf{V}_2^\top$ (Figure 4**C**).

Impact of task alignment on neural representations. From Figure 4**A**, we see that, relative to the single-task case, the task-specific output RSMs $\mathbf{W}_i\mathbf{W}_i^\top$ undergo a redistribution of correlations for fully aligned (a) and conflicting (b) tasks: the diagonal entries weaken ($\mathbf{\Delta USU}_{i,i}^\top < 0$), as the shared modes divide each task's self-correlation between the two tasks, while the cross-task entries become positive ($\mathbf{\Delta USU}_{i,j}^\top > 0$, $i \neq j$), reflecting new correlations between the tasks' outputs. For orthogonal tasks (c), both are unchanged ($\mathbf{\Delta USU}^\top = 0$). Similarly, Figure 4**C** shows that the shared input representation $\mathbf{W}_S^\top\mathbf{W}_S$ develops new or stronger structural relationships ($\mathbf{\Delta VSV}^\top \neq 0$) for aligned (a) and conflicting (b) tasks, relative to the direct sum of the single-task input RSMs (Equation 19). For orthogonal tasks (c), the combined input RSM is exactly this direct sum ($\mathbf{\Delta VSV}^\top = 0$), indicating no new structural relationships.

Impact of magnitude imbalance. Comparing Figures 4**A** and 4**B** shows that magnitude imbalance further reshapes the task-specific representations of aligned (c)-(d) and conflicting (e)-(f) tasks. When task 2 has a larger magnitude than task 1 ($\Delta M_{12} < 1$, Equation 12), the task-1 block of the joint RSM (top-left) is more strongly attenuated than in the balanced case (Figure 4**A**), even though task 1's own single-task representation is unchanged: the stronger task draws more of the shared modes, weakening the weaker task's block of the combined representation. For orthogonal tasks (d)-(h), this block is invariant to magnitude, since the tasks share no modes.

Our analysis shows that the analytical solutions in Theorem 3.2 determine the representations a linear MTL network learns at convergence, directly from the task correlation matrices. Joint training reshapes the representations of aligned and conflicting tasks relative to single-task learning, while orthogonal tasks are left unchanged, with the strength of these effects set by task alignment and magnitude imbalance.

6 Applications to ReLU Networks and Multi-MNIST

We empirically validate the theoretical predictions from Sections 4 and 5 using two-layer linear and ReLU networks trained on two variants of Multi-MNIST, a multi-task extension of MNIST (Lecun et al., 1998) introduced by Sabour et al. (2017). The dataset is constructed by overlaying pairs of MNIST digits, with left- and right-digit classification forming two separate tasks. To study the impact of task alignment, we consider the original dataset and a permuted variant that modifies task alignment, referred to as the *Baseline* and *Permuted* Multi-MNIST datasets, respectively. Additional details on the dataset can be found in Appendix D.3. Finally, to further demonstrate how our framework extends to high-dimensional settings, Appendix D.5 presents additional analysis on the NYUv2 (Silberman et al., 2012) dataset.

Alignment in Realistic Datasets. In realistic datasets, task relationships rarely fall cleanly into the fully aligned, conflicting, or orthogonal cases of Definition 2.1. Instead, the singular values α_i of $\mathbf{O}_{12}$ vary continuously between 1 and 0, mixing all three mode types within a single task pair. To illustrate how our analysis applies in this setting, we study two Multi-MNIST configurations, complemented by a synthetic analysis of mixed spectra in Appendix D.2. In the Baseline configuration, the spectrum mixes aligned and mildly conflicting directions ($\alpha_i \in [0.53, 1]$), with near-balanced task magnitudes ($\Delta M \approx 0.97$). In the Permuted configuration, a single direction remains fully aligned ($\alpha_1 = 1$) while the rest drop sharply into the

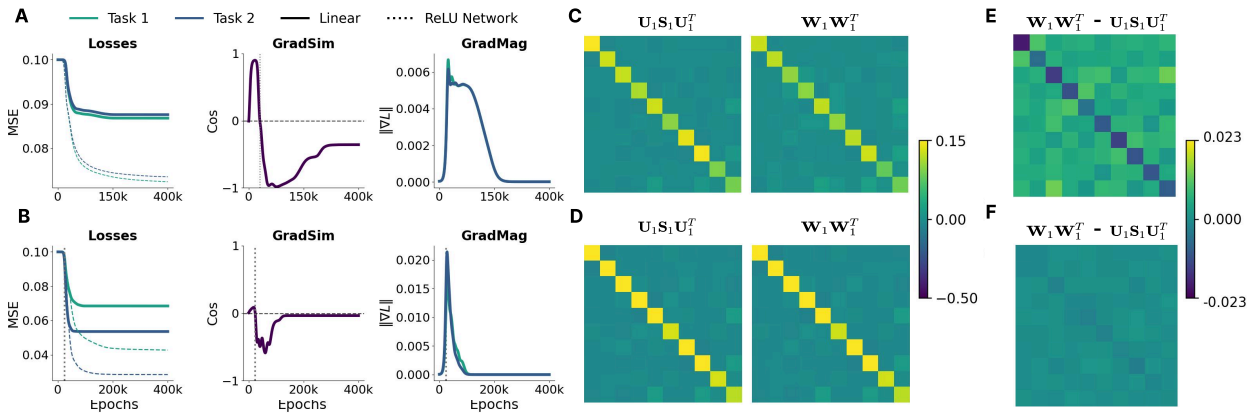

Figure 5: Learning dynamics and representations for two Multi-MNIST task pairs, for linear (solid) and ReLU (dotted) networks ($N_S = 20$, $\eta = 10^{-2}$, $\sigma_W = 10^{-4}$): Baseline (**A**, **C**, **E**), with ($\alpha_i \in [0.53, 1]$), and Permuted (**B**, **D**, **F**), with one aligned direction ($\alpha_1 = 1$) and nine strongly conflicting ones ($\alpha_i \leq 0.24$). **A, B:** losses, gradient cosine similarity, and gradient magnitude. Baseline converges nearly simultaneously with a sharp positive then deeply negative cosine similarity; Permuted shows one task converging faster, with a brief positive peak and a mild negative dip. **C-F:** STL RSMs versus MTL $\mathbf{W}_i\mathbf{W}_i^\top$ at convergence, with their difference in **E** (Baseline) and **F** (Permuted). Joint training reshapes the Baseline representation (weaker diagonal, stronger off-diagonal, **E**) but leaves Permuted close to STL (**F**).

strongly conflicting range ($0.14 \leq \alpha_i \leq 0.24$), yielding a pair that shares one common direction but conflicts along all others; here the task magnitudes are slightly imbalanced ($\Delta M \approx 0.82$).

Learning Dynamics. Figure 5(**A**-**B**) shows the learning dynamics of a linear and ReLU MTL network on the Baseline (**A**) and Permuted (**B**) Multi-MNIST datasets. For the Baseline task pair (Figure 5**A**), both tasks converge at a similar rate, and the cosine similarity rises sharply before becoming deeply negative. This is consistent with how the combined spectrum splits under this mixture (Appendix D.2): the aligned directions drive the early positive cooperation, while the mildly conflicting directions produce weak modes ($s_\xi = s\sqrt{1-\alpha}$) that resolve slowly, dominating the late-stage gradient (the deep negative similarity) and making the Baseline tasks converge more slowly than the Permuted. For the Permuted task pair (Figure 5**B**), only a single direction is aligned ($\alpha_1 = 1$), consistent with a brief early rise in cosine similarity and a short joint decrease in loss, before the strongly conflicting bulk of the spectrum ($\alpha_i \leq 0.24$) drives the cosine similarity moderately negative. The two tasks also converge at slightly different rates. In both cases, the linear and ReLU networks show similar loss trajectories.

Convergence Behaviour. In Figure 5(**C**-**D**), we compare the task-specific representation of the trained MTL network, $\mathbf{W}_1\mathbf{W}_1^\top$ at convergence, with the single-task RSM $\mathbf{U}_1\mathbf{S}_1\mathbf{U}_1^\top$ obtained directly from the data correlations (the representation an STL network would converge to, Section 5). After training on the Baseline tasks (Figure 5**C**), the MTL network reshapes the representation relative to this single-task prediction: the diagonal (self-)correlations weaken ($\mathbf{\Delta USU}^\top_{i,i} < 0$) while the off-diagonal (cross-task) correlations strengthen ($\mathbf{\Delta USU}^\top_{i,j} > 0$, $i \neq j$), as predicted across the aligned and conflicting regimes. For the Permuted tasks (Figure 5**D**), $\mathbf{W}_1\mathbf{W}_1^\top$ stays close to the single-task prediction ($\mathbf{\Delta USU}^\top \approx 0$), consistent with its weakly aligned spectrum ($\alpha_i \leq 0.24$): the tasks share little structure, so joint training barely reshapes the representation. Both effects are reproducible across seeds (Appendix D.3).

7 Discussion

By analysing the learning dynamics obtained from our Riccati-based analytical solution, we show how different regimes of task alignment drive qualitatively different interactions between tasks, and how these

interactions evolve over the course of training. Together with magnitude imbalance, alignment governs both the relative learning speeds of the tasks and the structural relationships reflected in the neural representations of the shared and task-specific layers. Although our analysis relies on simplifying assumptions, including linearity and two-layer depth, we validate our predictions with experiments on deeper architectures (Appendix D.3), ReLU networks, and the Multi-MNIST dataset. These findings offer a data-driven perspective, grounded in task alignment and magnitude imbalance, that motivates principled approaches to task grouping and dynamic loss weighting.

Practical Implications. Regarding task grouping, our results suggest that task alignment, estimable directly from data, characterizes both the feasibility of inter-task transfer (Wu et al., 2020) and the nature of task interactions throughout training (Sections 4 and 5). This points toward data-estimated alignment as a candidate signal for task grouping, complementing approaches that require iterative full-model training (Fifty et al., 2021; Sherif et al., 2024). Beyond task grouping, our results suggest that the benefits of dynamic loss weighting depend on the alignment and magnitude imbalance in the data. The degree of alignment sets not just whether tasks interfere but the character of their interaction over training and how it evolves across the different stages of learning. Because inter-task interference is thus inherently dynamic and data-dependent, effective weighting strategies should adapt to task alignment and magnitude imbalance rather than treat tasks in a data-agnostic manner, and dynamic weighting should not be assumed uniformly beneficial regardless of task structure.

Limitations & Future Work. Our analysis is restricted to deep linear networks under simplifying assumptions (A1-A3), which may not fully capture nonlinear feature learning or the representational complexity of real-world architectures. In Appendices C.6 and C.7, we show that the framework extends beyond its original formulation by relaxing Assumptions A2 and A3, accommodating both $\mathbf{\Gamma}$-balanced initializations and bottlenecked architectures. The assumptions of whitened inputs and linearity, however, remain central to the current theory. Prior work suggests that insights from linear models often transfer meaningfully to nonlinear settings (Nam et al., 2025; Pennington et al., 2017; Arora et al., 2018), and our empirical results on deeper architectures and more complex datasets provide further support for this. Extending the theory to nonlinear networks and more realistic data distributions therefore remains an important direction for future work. Beyond architectural generalization, an important next step is to establish a direct connection between task alignment and the generalization gains achievable through dynamic weighting, thereby linking the learning dynamics characterized here to downstream performance improvements.

Reproducibility. To facilitate reproducibility, we release the full source code, example Slurm scripts, and a notebook for reproducing all figures. We additionally provide the data used to generate the Multi-MNIST results, along with additional experimental runs. Appendix E contains detailed information about how each Figure in the main paper has been created.

Broader Impact Statement

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

References

Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.08.022. URL `https://www.sciencedirect.com/science/article/pii/S0893608020303117`.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 244–253. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/arora18a.html`.

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=KLmWRMg6nL.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, March 2000. ISSN 1076-9757. doi: 10.1613/jair.731. URL http://dx.doi.org/10.1613/jair.731.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pp. 567–580. Springer, 2003.

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, pp. 6615–6629. Curran Associates, Inc., 2022.

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

Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview. *ACM Computing Surveys*, 56(12):1–32, 2024.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Liam Collins, Hamed Hassani, Mahdi Soltanolkotabi, Aryan Mokhtari, and Sanjay Shakkottai. Provable multi-task representation learning by two-layer relu neural networks. *Proceedings of machine learning research*, 235:9292, 2024.

Sander Dalm, Joshua Offergeld, Nasir Ahmad, and Marcel van Gerven. Efficient deep learning with decorrelated backpropagation, 2025. URL https://arxiv.org/abs/2405.02385.

Clémentine Carla Juliette Dominé. Exact learning dynamics of bottlenecked and wide deep linear networks. In *UniReps: 3rd Edition of the Workshop on Unifying Representations in Neural Models*, 2025. URL https://openreview.net/forum?id=ssamp3oamY.

Clémentine Carla Juliette Dominé, Nicolas Anguita, Alexandra Maria Proca, Lukas Braun, Daniel Kunin, Pedro A. M. Mediano, and Andrew M Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZXaocmXc6d.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2020. URL https://arxiv.org/abs/1812.02224.

Cathrin Elich, Lukas Kirchdorfer, Jan M Köhler, and Lukas Schott. Examining common paradigms in multi-task learning. *arXiv preprint arXiv:2311.04698*, 2023.

Andreas Engel. *Statistical mechanics of learning*. Cambridge University Press, 2001.

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

Kenji Fukumizu. Effect of batch learning in multilayer neural networks. *Gen*, 1(04):1E–03, 1998.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f39ae9ff3a81f499230c4126e01f421b-Paper.pdf.

Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/cab070d53bd0d200746fb852a922064a-Paper.pdf.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Emilie Grégoire, Muhammad Hafeez Chaudhary, and Sam Verboven. Sample-level weighting for multi-task learning with auxiliary tasks. *Applied Intelligence*, 54(4):3482–3501, Feb 2024. ISSN 1573-7497. doi: 10.1007/s10489-024-05300-9. URL https://doi.org/10.1007/s10489-024-05300-9.

Yihang Guo, Tianyuan Yu, Liang Bai, Yanming Guo, Yirun Ruan, William Li, and Weishi Zheng. Revisit the imbalance optimization in multi-task learning: An experimental analysis, 2025. URL https://arxiv.org/abs/2509.23915.

Jinmin He, Kai Li, Yifan Zang, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Efficient multi-task reinforcement learning with cross-task policy guidance. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 117997–118024. Curran Associates, Inc., 2024a.

Yifei He, Shiji Zhou, Guojun Zhang, Hyokun Yun, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Han Zhao. Robust multi-task learning with excess risks. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=JzWFmMySpn.

Devon Jarvis, Sebastian Lee, Clémentine Carla Juliette Dominé, Andrew M Saxe, and Stefano Sarao Mannelli. A theory of initialisation's impact on specialisation*. *Journal of Statistical Mechanics: Theory and Experiment*, 2025(11):114001, nov 2025. doi: 10.1088/1742-5468/ae1214. URL https://doi.org/10.1088/1742-5468/ae1214.

Daniel Kunin, Allan Raventos, Clémentine Carla Juliette Dominé, Feng Chen, David Klindt, Andrew M Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eNM94i7R3A.

Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and M. Pawan Kumar. In defense of the unitary scalarization for deep multi-task learning. *CoRR*, abs/2201.04122, 2022. URL https://arxiv.org/abs/2201.04122.

Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *Published as a conference paper at ICLR 2019*, 2018.

Isabelle Leang, Ganesh Sistu, Fabian Bürger, Andrei Bursuc, and Senthil Yogamani. Dynamic task weighting methods for multi-task networks in autonomous driving systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, 2020. doi: 10.1109/ITSC45102.2020.9294676.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Sebastian Lee, Sebastian Goldt, and Andrew Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pp. 6109–6119. PMLR, 2021.

Baijiong Lin and Yu Zhang. LibMTL: A Python library for multi-task learning. *Journal of Machine Learning Research*, 24(209):1–7, 2023.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jjtFD8A1Wx.

Samuel Lippl and Jack Lindsey. Inductive biases of multi-task learning and finetuning: multiple regimes of feature reuse. *Advances in Neural Information Processing Systems*, 37:118745–118776, 2024.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d27fdf2477ffbff837d73ef7ae23db9-Paper.pdf.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 57226–57243. Curran Associates, Inc., 2023.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Yoonsoo Nam, Seok Hyeong Lee, Clémentine Carla Juliette Dominé, Yeachan Park, Charles London, Wonyl Choi, Niclas Alexander Göring, and Seungjai Lee. Position: Solve layerwise linear models first to understand neural dynamical phenomena (Neural collapse, emergence, Lazy/Rich regime, and grokking). In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81897–81929. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/nam25a.html.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/doi/abs/10.1073/pnas.2015509117.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

Hannah Pinson, Joeri Lenaerts, and Vincent Ginis. Linear cnns discover the statistical structure of the dataset using only the most dominant frequencies. *Proceedings of the 40th International Conference on Machine Learning*, 202:27876–27906, 2023. URL https://icml.cc/Conferences/2023.

Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. URL https://arxiv.org/abs/1706.05098.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3859–3869, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ashwin Saraswatula and David Klindt. Data whitening improves sparse autoencoder learning, 2025. URL https://arxiv.org/abs/2511.13981.

Tetsushi Sasagawa. On the finite escape phenomena for matrix riccati equations. *IEEE Transactions on Automatic Control*, 27(4):977–979, 1982.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The neural race reduction: Dynamics of abstraction in gated networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19287–19309. PMLR, 17–23 Jul 2022.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *2nd International Conference on Learning Representations, 2014.*, 2014.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1820226116.

H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Phys. Rev. A*, 45:6056–6091, Apr 1992. doi: 10.1103/PhysRevA.45.6056. URL https://link.aps.org/doi/10.1103/PhysRevA.45.6056.

Ammar Sherif, Abubakar Abid, Mustafa Elattar, and Mohamed ElHelw. Stg-mtl: scalable task grouping for multi-task learning using data maps. *Machine Learning: Science and Technology*, 5(2):025068, June 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad4e04. URL http://dx.doi.org/10.1088/2632-2153/ad4e04.

Nathan Silberman, Hoiem Derek, Kohli Pushmeet, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pp. 9767–9779. PMLR, 2021.

Xiaozhuang Song, Shun Zheng, Wei Cao, James Yu, and Jiang Bian. Efficient and effective multi-task grouping via meta learning on task combinations. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37647–37659. Curran Associates, Inc., 2022.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pp. 9120–9132. PMLR, 2020.

Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10153–10161. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/tarmoun21a.html.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. Do current multi-task optimization methods in deep learning even help? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, pp. 13597–13609. Curran Associates, Inc.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(2):35–63, 2007. URL http://jmlr.org/papers/v8/xue07a.html.

Wei-Yong Yan, U. Helmke, and J.B. Moore. Global analysis of oja's flow for neural networks. *IEEE Transactions on Neural Networks*, 5(5):674–683, 1994. doi: 10.1109/72.317720.

Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14071–14081, 2023.

Yedi Zhang, Peter E Latham, and Andrew M Saxe. Understanding unimodal bias in multimodal deep linear networks. In *Forty-first International Conference on Machine Learning*, 2024.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

Yu Zhang and Dit-Yan Yeung. A regularization approach to learning task relationships in multitask learning. *ACM Trans. Knowl. Discov. Data*, 8(3), June 2014. ISSN 1556-4681. doi: 10.1145/2538028. URL `https://doi.org/10.1145/2538028`.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3239–3248, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.296. URL `https://aclanthology.org/2020.acl-main.296/`.

Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 817–824, 2019.

A Additional Related Work

In this section, we review additional related work in theoretical single-task and multi-task learning and compare it to our contributions.

A.1 Learning Dynamics of Deep Linear Networks

Our work extends several foundational theoretical analyses of the continuous-time limit of gradient descent, known as *gradient flow*, to the setting where multiple tasks are learned simultaneously, i.e., multi-task learning (MTL). We build on prior theoretical studies of deep linear networks, including those by Saxe et al. (2014), Fukumizu (1998), and Braun et al. (2022). As a key starting point, Saxe et al. (2014) derived nonlinear, coupled differential equations that describe the learning dynamics of network weights in terms of input-output statistics. They provided exact analytical solutions under *tabula rasa* (i.e., small random weights) initializations and whitened input data. Later work (Saxe et al., 2019) extended these results to both shallow and deep networks, offering insights into the semantic development of representations in terms of effective singular values and the temporal evolution of hidden layers. However, whereas their analysis focuses on how different classes are learned over time within a shared multi-output setting, our framework addresses a distinct scenario: multiple tasks are learned in parallel, each with its task-specific gradients and dedicated layers. While the analytical expressions in Saxe et al. (2014; 2019) rely on the assumptions of linear networks, specific initialization schemes, and whitened inputs, Pennington et al. (2017) demonstrated how theoretical insights from linear networks can inform improved initialization strategies for nonlinear models. Similarly, Gidel et al. (2019) extended aspects of the gradient flow analysis to discrete-time gradient descent, narrowing the gap between theory and practice. As such, these theoretical insights are still believed to offer important insights for more general architectures and initializations.

Building on the foundational analyses from Saxe et al. (2014), subsequent work Lampinen & Ganguli (2018); Advani et al. (2020); Goldt et al. (2019) developed an analytical theory for generalization in deep linear neural networks in a teacher-student setup (Seung et al., 1992; Engel, 2001). This setup has also been applied to analyse continual learning (Lee et al., 2021) (including the problem of *catastrophic forgetting* (Goodfellow et al., 2013)) and transfer learning (Lampinen & Ganguli, 2018). Additionally, Saxe et al. (2022) introduced the Gated Deep Linear Network (GDLN) framework to allow for non-linear functions of inputs to learn more about the implications of architectural choices on the learning dynamics for spectral initializations. While they allow for output-specific nodes, they only consider models updated with one overall gradient and no task-specific weights. Finally, Zhang et al. (2024) analyses the dynamics of multi-modal learning, where different modalities (different inputs) are used to learn a common task (same output). They obtain

analytical solutions for uncorrelated modalities and analyse unimodal bias for early and late fusion networks through their loss landscape. However, their setup should not be confused with our work, which considers multiple outputs from a similar input.

Our work is mostly related to the recent line of work Tarmoun et al. (2021); Braun et al. (2022); Dominé et al. (2025) that builds from the matrix Riccati formulation of gradient descent (Fukumizu, 1998) and matrix factorization. In Tarmoun et al. (2021); Braun et al. (2022), gradient flow equations are written for single-task learning as matrix Riccati equations, which have finite solutions under some assumptions (Sasagawa, 1982) that hold for a richer class of initial conditions than the *Tabula Rasa initialization* as encountered in the analytical solutions obtained in Saxe et al. (2014) and subsequent work. In particular, Braun et al. (2022) uses this formalism and describes alignment dynamics and convergence behaviour that do not occur in the tabula rasa setting for zero-balanced weight initialization, corresponding to rich dynamics, and Dominé et al. (2025) extends the obtained solution to more general non-zero-balanced initializations. Both Braun et al. (2022); Dominé et al. (2025) apply their formalism to transfer and/or continual learning, where different tasks are learned sequentially, but do not mention multi-task learning, where tasks are learned simultaneously.

A.2 Theoretical Work on Multi-Task learning

Early theoretical work in multi-task learning (Ben-David & Schuller, 2003; Xue et al., 2007) focused on Kernel Learning and states that multi-task works well on similar tasks, where the notion of similarity is quantified based on the similarity of the corresponding single-task models (e.g. decision boundaries are close). In Wu et al. (2020), this notion of similarity is extended to neural networks where task data similarity is considered as a second-order factor that impacts interference between multiple tasks in hard-parameter sharing MTL networks. Based on that observation, they derive bounds for the maximal capacity of the shared layer and the number of datapoints of the source task needed to enforce interference between tasks. Accordingly, they provide a static, geometry-based characterization of task similarity and transfer conditions in MTL neural networks, without accounting for the temporal dynamics or describing how these interactions evolve. The main findings of Wu et al. (2020) are: a too high capacity of the shared module eliminates any kind of information transfer between tasks, changing the number of data points, and a notion of STL covariance similarity impacts the amount of transfer between tasks. In contrast to their work, we show that, next to the training losses, transfer can be studied by considering the representations and output functions. Furthermore, this work considers how the evolution of the differences in alignment affects the learning dynamics (i.e., the learning speed and internal representations) while Wu et al. (2020) defines transfer as a static notion related to performance on the validation set.

Particularly relevant to our work is the analysis presented in Lampinen & Ganguli (2018), which introduced the concept of task alignment through a similarity matrix constructed from the right singular vectors associated with different tasks. Their study is situated within a student-teacher framework and focuses on the derivation of generalization dynamics in single-task networks. By deriving analytical expressions for the generalization error at early stopping, they define a measure of transfer benefit as the difference in generalization error when tasks are trained jointly versus separately. In the specific setting of two low-rank teacher models, they demonstrate that transfer in multi-task learning is primarily determined by the initial alignment between tasks and the magnitudes of their respective singular values, while being independent of the output singular vectors. Building on this work, we also utilize the similarity matrix to define various classes of alignment. However, rather than focusing on generalization error, our analysis is concerned with how task alignment shapes the learning dynamics in a way that is independent of the student-teacher paradigm and accommodates more general initializations and model architectures.

While Lampinen & Ganguli (2018) and Wu et al. (2020) primarily quantify transfer in terms of its effect on a static generalization, our work goes further by characterizing both how transfer evolves throughout training and what forms it can take. Specifically, we examine not only how tasks influence each other's final performance, but also how they affect the trajectory of learning, the speed of convergence, and the structure of intermediate representations. In contrast to prior analyses, we aim to provide a more detailed account of the dynamics of inter-task interaction, offering a richer understanding of transfer beyond end-point generalization metrics.

A.3 Practical algorithms for task grouping and dynamic task loss weighting

Recent work in MTL is often focused on the design of dynamic weighting algorithms with the goal of eliminating negative task interference/transfer (He et al., 2024b; Guo et al., 2025). At the base of the design of these different algorithms are some common paradigms concerning the source of negative interference, i.e., *gradient imbalance* (Chen et al., 2018; Leang et al., 2020) which occurs when the gradients of different tasks have large differences in magnitudes and *conflicting gradient directions* (Du et al., 2020; Liu et al., 2021; 2023; Ban & Ji, 2024) which occurs when task-specific gradients have negative cosine similarities. Recently, these paradigms have been questioned and critically analysed (Guo et al., 2025; Elich et al., 2023; Xin et al.). To address the mentioned problems, we aim to understand the learning dynamics and propose a theoretical framework to effectively study task interactions from a new perspective.

Critical Studies on Dynamic Weighting Algorithms. While it is clear that in many cases, dynamic weighting algorithms outperform Equal Weighting algorithms where tasks are given the same static, task-specific weight, (Lin et al., 2022) showed that random dynamical weighting yields similar performance to computationally expensive SOTA dynamic weighting algorithms designed for task-conflicting heuristics. Similarly, Xin et al.; Kurin et al. (2022) performed an extensive empirical search and found that most of the performance differences between MTL models are much more sensitive to the choice of hyperparameters than to the choice of dynamical weighting algorithms, highlighting the point made by Elich et al. (2023) that a better theoretical understanding is necessary to complement the popular gradient based task-weighting algorithms and rethink the common paradigms in MTL literature. According to Grégoire et al. (2024), a necessary step is to disentangle interference that is not related to differences between tasks as a whole with interference coming from different samples within the same task, since gradient interference is not necessarily unique to MTL setups (Elich et al., 2023).

Task Grouping. Another line of research in MTL literature is concerned with the fundamental question of which tasks should be learned together in a multi-task model to maximize cooperation and minimize competition. Since the search space of possible task groupings grows exponentially with the number of tasks, exhaustive evaluation of different groups is infeasible even for moderate task counts. Recent works have approached this challenge from several angles, e.g., regularization-based methods that learn task relationships as part of the training objective (Zhang & Yeung, 2014) heuristic or empirical methods that evaluate candidate groupings (often via training on subsets or proxy metrics) (Standley et al., 2020; Sherif et al., 2024), and meta-learning approaches that predict optimal groupings or task affinities (Song et al., 2022; Fifty et al., 2021).

B Multi-Task Learning: Preliminaries

This work considers a Multi-Task Learning (MTL) (Ruder, 2017) setup in which the objective is to simultaneously learn N_T related tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_{N_T}\}$. All tasks share a common input vector $\mathbf{x} \in \mathbb{R}^{N_1}$, where N_1 is the input dimensionality, but each task $\mathcal{T}_i$ is associated with task-specific output vectors $\mathbf{y}_i \in \mathbb{R}^{N_{io(l)}}$, where N_{io} denotes the output dimensionality of task $\mathcal{T}_i$. To this end, a standard hard-parameter sharing architecture is adopted Zhang & Yang (2021). In this setup, the tasks are trained jointly using shared weight matrices $\mathbf{W}_S = \prod_{\ell=1}^{L_S} \mathbf{W}_S^{(\ell)}$followed by $\mathcal{T}_i$-specific weight matrices $\mathbf{W}_i = \prod_{\ell=1}^{L_i} \mathbf{W}_i^{(\ell)}$. The difference with a single-task two-layer linear network is visualised in Figure 1.

The MTL network is trained by minimizing a composite loss function $\mathcal{L}_{\text{Tot}}$, defined as a weighted sum of task-specific losses $\mathcal{L}_i$,

$$\mathcal{L}_{\text{Tot}}(t) = \sum_{i=1}^{N_T} \omega_i(t) \mathcal{L}_i(\mathbf{W}_i, \mathbf{W}_S, t), \tag{21}$$

where $\omega_i(t)$ denotes the time-dependent loss weighting coefficient for task $\mathcal{T}_i$, controlling its relative contribution to the overall loss at training iteration t. Here, t corresponds to the number of epochs. We assume uniform static weighting $\omega_i = 1$ but discuss the impact of $\omega_i \neq 1$ in Appendix D.

In this work, the neural network is trained with full-batch gradient descent with a mean squared error (MSE) task-specific loss function $\mathcal{L}_i^{\text{MSE}}$ for each task,

$$\mathcal{L}_i^{\text{MSE}} = \frac{1}{2N} \sum_{\mu=1}^{N_D} ||\mathbf{y}_i^\mu - \mathbf{W}_i \mathbf{W}_S \mathbf{x}^\mu| \,|^2, \tag{22}$$

where N_D is the number of input samples and $\mathbf{x}^\mu$, $\mathbf{y}_i^\mu$ denote the μ-th input and corresponding target output for task $\mathcal{T}_i$. While the shared weights $\mathbf{W}_S$ are updated based on the gradient computed on the total loss $\mathcal{L}_{\text{Tot}}(t)$ (equation 21), the task-specific weights are updated through task-specific gradients computed on the task-specific losses $\mathcal{L}_i$ and learning rate η according to the update rule,

$$\mathbf{W}_S^\ell(t+1) \leftarrow \mathbf{W}_S^\ell(t) - \eta \nabla_{\mathbf{W}_S^\ell(t)} \mathcal{L}_{Tot}(t), \qquad \mathbf{W}_i^\ell(t+1) \leftarrow \mathbf{W}_i^\ell(t) - \eta \nabla_{\mathbf{W}_i^\ell(t)} \mathcal{L}_i(t), \tag{23}$$

such that $\forall i \neq j : \nabla_{\mathbf{W}_i} \mathcal{L}_j = 0$.

B.1 Two-Layer Linear Two-Task networks

In line with prior studies on the learning dynamics in single-task learning, we consider the setup of a two-layer linear network. Specifically, we use MTL networks with hard-parameter sharing (Ruder, 2017) where the shared weights $\mathbf{W}_S$ are updated through the gradient computed on the composite loss of both tasks, $\mathcal{L}_{MTL}$,

$$\begin{aligned} \mathbf{W}_S^{(t+1)} &= \mathbf{W}_S^{(t)} - \eta \nabla_{\mathbf{W}_S^{(t)}} \mathcal{L}_{MSE} \\ &= \mathbf{W}_S^{(t)} - \frac{1}{N_D} \eta \nabla_{\mathbf{W}_S^{(t)}} \sum_i^{N_T} \sum_{\mu=1}^{2N_D} \|\mathbf{y}_i^\mu - \mathbf{W}_i \mathbf{W}_S \mathbf{x}^\mu\|^2, \end{aligned} \tag{24}$$

The gradient with respect the shared layer $\mathbf{W}_S$, is given by

$$\begin{aligned} \nabla_{\mathbf{W}_S^{(t)}} \mathcal{L}_{MSE} &= -\sum_{i=1}^{N_T} \sum_{\mu=1}^{N_D} \mathbf{W}_i^T \left(\mathbf{y}_i^\mu - \mathbf{W}_i \mathbf{W}_S \mathbf{x}^\mu\right) \mathbf{x}^{\mu T} \\ &= -\sum_{i=1}^{N_T} \sum_{\mu=1}^{N_D} \left(\mathbf{W}_i\right)^\top \left(\mathbf{y}_i^\mu \mathbf{x}^{\mu T} - \mathbf{W}_i \mathbf{W}_S \mathbf{x}^\mu \mathbf{x}^{\mu T}\right) \end{aligned} \tag{25}$$

In the limit of small learning rates $\eta \to 0$, the updates of the weights are sufficiently small such that $\Delta W^\top \to dW^\top$ and the gradient descent dynamics are well approximated by the continuous time differential equation (Saxe et al., 2014),

$$\tau \frac{d\mathbf{W}_S}{dt} = \sum_{i=1}^{N_T} \mathbf{W}_i^\top (\mathbf{\Sigma}_{io} - \mathbf{W}_i \mathbf{W}_S \mathbf{\Sigma}_{11}), \tag{26}$$

with $\tau = \frac{1}{\eta}$, which we refer to as *Gradient Flow*. Where we defined the input-input and input-output correlation matrices as,

$$\begin{aligned} \mathbf{\Sigma}_{11} &\equiv \frac{1}{N_D} \sum_{\mu=1}^{N_D} \mathbf{x}^\mu (\mathbf{x}^\mu)^\top \\ \mathbf{\Sigma}_{io} &\equiv \frac{1}{N_D} \sum_{\mu=1}^{N_D} \mathbf{y}_i^\mu (\mathbf{x}^\mu)^\top. \end{aligned} \tag{27}$$

In contrast, the task-specific gradient signals are explicitly separated for the task-specific weights $\mathbf{W}_i$, which get updated through,

$$\mathbf{W}_i^{(t+1)} = \mathbf{W}_i^{(t)} - \frac{1}{N_D} \eta \nabla_{\mathbf{W}_i^{(t)}} \sum_{\mu=1}^{N_D} \|\mathbf{y}_i^\mu - \mathbf{W}_i \mathbf{W}_S \mathbf{x}^\mu\|^2 \tag{28}$$

yielding,

$$\tau \frac{d\mathbf{W}_i}{dt} = \left(\mathbf{\Sigma}_{io} - \mathbf{W}_i \mathbf{W}_S \mathbf{\Sigma}_{11}\right) \mathbf{W}_S^\top, \tag{29}$$

in the continuous limit. Assuming whitened inputs ($\mathbf{\Sigma}_{11} = \mathbf{I}$), the gradient-flow equations for learning two tasks simultaneously are

$$\tau \frac{d\mathbf{W}_S}{dt} = \mathbf{W}_1^\top (\mathbf{\Sigma}_{1o} - \mathbf{W}_1 \mathbf{W}_S) + \mathbf{W}_2^\top (\mathbf{\Sigma}_{2o} - \mathbf{W}_2 \mathbf{W}_S), \tag{30}$$

$$\tau \frac{d\mathbf{W}_1}{dt} = \left(\mathbf{\Sigma}_{1o} - \mathbf{W}_1 \mathbf{W}_S\right) \mathbf{W}_S^\top, \qquad \tau \frac{d\mathbf{W}_2}{dt} = \left(\mathbf{\Sigma}_{2o} - \mathbf{W}_2 \mathbf{W}_S\right) \mathbf{W}_S^\top. \tag{31}$$

For $N_T > 2$ the derivation is completely analogous. These equations can be written compactly by stacking the task labels and task-specific weights,

$$\mathbf{Y}_{MTL} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix}, \qquad \mathbf{W}_{MTL} = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix}, \qquad \xi = \begin{pmatrix} \mathbf{\Sigma}_{1o} \\ \mathbf{\Sigma}_{2o} \end{pmatrix}, \tag{32}$$

giving

$$\tau \frac{d\mathbf{W}_S}{dt} = \mathbf{W}_{MTL}^\top (\xi - \mathbf{W}_{MTL} \mathbf{W}_S), \tag{33}$$

$$\tau \frac{d\mathbf{W}_i}{dt} = \left(\xi_{[r_i,:]} - \mathbf{W}_{MTL,[r_i,:]} \mathbf{W}_S\right) \mathbf{W}_S^\top, \tag{34}$$

where $r_i = \left[\sum_{j<i} N_{jo} : \sum_{j \leq i} N_{jo}\right]$ denotes the row block corresponding to task i (all columns). These are just like the single-task equations solved for a family of initializations in Saxe et al. (2014). In the main paper, we discuss how the solutions to these equations serve as a way to describe inter-task dynamics. In Appendix B.3 we discuss solutions from *Tabula Rasa* initializations and the special case of similar statistical tasks. In Appendix C we discuss how to rewrite these equations into the Riccati formalism which enables us to find an expression for the output functions and internal representations for a broader family of initializations as compared to the Tabula Rasa case.

B.2 Differences and Application to Single-Task Learning

In contrast to single-task learning (STL), our framework accommodates heterogeneous tasks (e.g., semantic segmentation and depth estimation (Liu et al., 2019)) with task-specific losses and gradient updates that separate the task-signals in parts of the network. Furthermore, each task's influence can be controlled through task-specific loss weights (equation 21), while supporting both shared and task-specific layers. Certain single-task problems, such as multi-class classification, can easily be reframed as multi-task problems by defining subgroups of tasks and task-specific layers. As such, our analysis can also shed light on how class-specific modes interact during training, and compare it to learning each class/subgroup of tasks in a separate model. Moreover, this architecture allows for different weighting between these groups or a different kind of activation/loss function, which will typically occur when combining regression and classification tasks (e.g., simultaneous depth and segmentation tasks (Liu et al., 2021)). As such, we can see this work as a generalization of earlier solutions proposed in (Braun et al., 2022) but with new insights concerning the alignment of subgroups of tasks.

B.3 Solving the equations in the Total MTL space: Tabula Rasa

In this section, we extend the methodology proposed in (Saxe et al., 2014; 2019) to implement the notion of multiple tasks into the analytical solutions to the learning dynamics of gradient flow. Decomposing $\xi = \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^\top$ and performing the change of variables $\mathbf{W}_S = \bar{\mathbf{W}}_S \mathbf{V}_\xi^\top$ and $\mathbf{W}_{MTL} = \mathbf{U}_\xi \bar{\mathbf{W}}_{MTL}$ yields the following equations,

$$\tau \frac{d\bar{\mathbf{W}}_S}{dt} = \bar{\mathbf{W}}_{MTL}^\top (\mathbf{S}_\xi - \bar{\mathbf{W}}_{MTL} \bar{\mathbf{W}}_S) \tag{35}$$

$$\tau \frac{d\bar{\mathbf{W}}_{MTL}}{dt} = (\mathbf{S}_\xi - \bar{\mathbf{W}}_{MTL} \bar{\mathbf{W}}_S) \bar{\mathbf{W}}_S^\top. \tag{36}$$

Hence, these formally look like the STL equations, which are solved in (Saxe et al., 2014) for a special class of initializations (often referred to as spectral or Tabula Rasa (random and small) initializations). Unlike STL analyses, which treat each task's gradient flow in isolation, our approach solves a single coupled dynamical system where one shared hidden matrix is pulled simultaneously by the residuals of all tasks, letting us capture transfer and interference effects that STL cannot reveal. By diagonalizing the joint task space and deriving a closed-form solution to the dynamics, we obtain an expression of what every task shares and the task-specific read-outs which represent how much each task retracts from a certain mode. To recover the latter, we construct,

$$\mathbf{U}_\xi = \begin{pmatrix}\mathbf{U}_{\xi 1} & \mathbf{U}_{\xi 2}\end{pmatrix}^\top, \tag{37}$$

where $\mathbf{U}_{\xi 1}$ and $\mathbf{U}_{\xi 2}$ are obtained by slicing the correct number of rows for task 1 and task 2. As such, the task-specific output maps are given by,

$$\mathbf{W}_1\mathbf{W}_S = \mathbf{U}_1\mathbf{H}(t)\mathbf{V}_\xi \qquad \mathbf{W}_2\mathbf{W}_S = \mathbf{U}_2\mathbf{H}(t)\mathbf{V}_\xi, \tag{38}$$

where $\mathbf{H}(\mathbf{t})$ is the time-dependent diagonal effective singular value matrix, with diagonal values h^α corresponding to a mode in the shared space ξ, which are found by solving equations (36),

$$h^\alpha(t) = \frac{e^{\frac{2s_\xi t}{\tau}} s_\xi}{\frac{s_\xi}{h_0^2} + e^{\frac{2s_\xi t}{\tau}} - 1}, \tag{39}$$

as shown in Saxe et al. (2014). Since the rank and the values of the singular value matrix $\mathbf{S}_\xi$ are determined by the alignment of the two tasks, the MTL learning dynamics are as well. While both tasks feel the same amplitude of the effective singular value h^α, they react to it in a different way, which is determined by the left singular values $\mathbf{U}_{\xi,i}$. The way these task-specific layers react to the same singular mode α is quantified through gamma, $\gamma_\alpha^{(j)}$,

$$\gamma_\alpha^{(j)} = \left\|u_\alpha^{(j)}\right\| = \sqrt{\sum_{k=1}^{N_{j,o}} \left[U_\xi\right]_{k,\alpha}^2}, \quad j \in \{1,2\}, \tag{40}$$

such that for for each mode α the task-specific "effective" singular values $a_\alpha(t)$, $b_\alpha(t)$ evolve according to,

$$a_\alpha(t) = \gamma_\alpha^{(1)}\, h_\alpha(t), \qquad b_\alpha(t) = \gamma_\alpha^{(2)}\, h_\alpha(t). \tag{41}$$

In equation (41), the scalars γ_i^j are constant prefactors that measure how strongly each task j "loads" onto mode-i, e.g., a mode with $\gamma_i^1 >> \gamma_i^2$ is mostly used by task one while a mode with $\gamma_i^1 \approx \gamma_i^2$ is shared by both tasks equally. Since both tasks feel the same amplitude of the effective singular value h^α, they read out the same modes at the same rate, determined by the task that perceives the highest strength for this mode.

B.4 Solving the equations for Tabula Rasa initializations

For completeness, we show how to obtain the solution in equation (39). The equations in (36) are typically solved for the columns of the synaptic matrices $\bar{\mathbf{W}}_S$ and $\bar{\mathbf{W}}_{MTL}$. Let h^α be the α-th column of the shared synaptic weight matrix $\bar{\mathbf{W}}_S$, representing a column vector of N_S synaptic weights connecting input mode α to the hidden layer, m^α a column vector of N_s synaptic weights connecting shared neurons to the ξ-output mode α. The equations (36) in terms of these columns vectors are given by,

$$\tau \frac{d}{dt} h^\alpha = [(s_\xi^\alpha - h^\alpha.m^{\alpha,\top})m^{\alpha,\top} - \sum_{\alpha\neq\gamma} m^{\gamma,\top}(h^\alpha.m^{\gamma,\top})] \tag{42}$$

$$\tau \frac{d}{dt} m^\alpha = (s_\xi^\alpha - h^\alpha.m^{\alpha,\top})m^\alpha - \sum_{\alpha\neq\gamma} h^\gamma(m^\alpha.m^\gamma). \tag{43}$$

To further simplify the vector equations (43), we assume a special class of initializations (i.e., Tabula Rasa) in which the equations the cross-mode contributions in (43) disappear. Namely, $h^\alpha, a^\alpha, b^\alpha \sim r^\alpha \; \forall\, \alpha \in \{1, ..., N_2\}$ such that $r^\alpha.r^\beta = \delta^{\alpha\beta}$ and all other connectivity modes set to zero. Under these assumptions, the modes

evolve independently from each other. Projecting everything along the r^α direction ($h^\alpha.r^\alpha = h, m^\alpha.r^\alpha = a, s^\alpha_\xi.r^\alpha = s_\xi$) turns the equations in (43) to the coupled scalar equations for each mode α,

$$\tau \frac{dh}{dt} = m(s_\xi - mh) \tag{44}$$

$$\tau \frac{da}{dt} = m(s_\xi - mh). \tag{45}$$

Since $h \sim m$ for our initializations, one can solve the equation (45) in terms of the variable $u = hm$, which allows us to capture the dynamics with one equation,

$$\tau \frac{du}{dt} = 2u(s_\xi - u), \tag{46}$$

which is separable and yields the following expression in terms of t, which can be reverted to obtain (39). In subsection B.5, we solve the equations (31) in another task basis for similar tasks. Note that zero-balanced initializations, as assumed in the main paper, are more general than the Tabula Rasa initialization assumed here. In Appendix C we discuss the derivation of the solutions under this more general assumption.

B.5 Solving the equations for similar statistical structure tasks: Tabula Rasa

To get a better understanding of the interpretation of the solution (39), we consider the case of tasks with a similar statistical structure, with similar and imbalanced singular values. To this end, we consider the singular value decomposition (SVD) of the first task's input-output correlation matrix,

$$\mathbf{\Sigma}_{1o} = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^\top, \tag{47}$$

where $\mathbf{V}_1$ is a $N_1 \times N_1$ orthogonal matrix whose columns contain input-analyzing singular vectors v_α that reflect independent modes of variation in the input, $\mathbf{U}_1$ is an $N_{31} \times N_{31}$ orthogonal matrix whose columns contain output-analyzing vectors u_α that reflect independent modes of variation in the output and $\mathbf{S}_1$ is an $N_{31} \times N_1$ diagonal matrix containing the singular values s^1_α, ordered such that $s^1_1 > s^1_2 ... > s_{N_1}$.

We transform the variables in equation (31) to the task 1 synaptic weight space according to $\mathbf{W}_S = \bar{\mathbf{W}}_S V_A^\top$, $\mathbf{W}_1 = \mathbf{U}_1 \bar{\mathbf{W}}_1$, $\mathbf{W}_2 = \mathbf{U}_1 \bar{\mathbf{W}}_2$ (where we assume both tasks have the same dimension for a moment so we can project $\mathbf{W}_2$ along the $\mathbf{U}_1$ basis). By studying the evolution of $\bar{\mathbf{W}}_2$ we learn how $\mathbf{W}_2$ evolves along the singular directions of task's 1 data structure. Note that we can find the original weights back once we found solutions for $\bar{\mathbf{W}}_i$ and $\bar{\mathbf{W}}_S$.

This transforms the equations (7) to,

$$\tau \dot{\bar{\mathbf{W}}}_S = \bar{\mathbf{W}}_1^\top (\mathbf{S}_1 - \bar{\mathbf{W}}_1 \bar{\mathbf{W}}_S) + \bar{\mathbf{W}}_2^T (\mathbf{U}_1^T \mathbf{\Sigma}_{2o} \mathbf{V}_1 - \bar{\mathbf{W}}_2 \bar{\mathbf{W}}_S) \tag{48}$$

$$= \bar{\mathbf{W}}_1^T (\mathbf{S}_1 - \bar{\mathbf{W}}_1 \bar{\mathbf{W}}_S) + \bar{\mathbf{W}}_2^T (\mathbf{S}_2^{proj} - \bar{\mathbf{W}}_2 \bar{\mathbf{W}}_S) \tag{49}$$

$$\tau \dot{\bar{\mathbf{W}}}_1 = (\mathbf{S}_1 - \bar{\mathbf{W}}_1 \bar{\mathbf{W}}_S) \bar{\mathbf{W}}_S^T \tag{50}$$

$$\tau \dot{\bar{\mathbf{W}}}_2 = (\mathbf{S}_2^{proj} - \bar{\mathbf{W}}_2 \bar{\mathbf{W}}_S) \bar{\mathbf{W}}_S^T, \tag{51}$$

where we defined $\mathbf{S}_2^{proj} \equiv \mathbf{U}_1^T \mathbf{\Sigma_{2o}} \mathbf{V}_1$, which is not necessary diagonal (in contrast to $\mathbf{S}_1$). $\mathbf{S}_2^{proj}$ quantifies how much the second task aligns with the latent structure extracted from the first task. If $\mathbf{S}_2^{proj}$ is nearly diagonal, it suggests that both tasks rely on similar input-output relationships. Conversely, if there is significant off-diagonal structure, it indicates that task 2 involves a different or more mixed combination of the underlying features captured by task 1. By studying the evolutions of the weights in the basis of the singular directions of task 1 we learn how this initial alignment or dis-alignment between the tasks evolves throughout training updates.

Assuming similar tasks, we obtain a nearly diagonal $\mathbf{S}_{proj}$. Similar to the total-space case, we solve the resulting equations for the columns of these synaptic matrices. Let h^α be the α-th column of the shared synaptic weight matrix, representing a column vector of N_2 synaptic weights connecting input mode α to

the hidden layer, a^α a column vector of N_2 synaptic weights connecting shared neurons to the task-1 output mode α and b^α as a vector containing weights which connect hidden layers to the task-2 output mode α as projected in the task-1 weight space. The equations (51) in terms of these columns vectors are given by,

$$\tau \frac{d}{dt} h^\alpha = [(s_1^\alpha - h^\alpha . a^{\alpha,\top}) a^{\alpha,\top} - \sum_{\alpha \neq \gamma} a^{\gamma,\top} (h^\alpha . a^{\gamma,\top})] \tag{52}$$

$$+ [(s_{2,proj}^\alpha - h^\alpha . b^{\alpha,\top}) b^{\alpha,\top} - \sum_{\alpha \neq \gamma} b^{\gamma,\top} (h^\alpha . b^{\gamma,\top})] \tag{53}$$

$$\tau \frac{d}{dt} a^\alpha = (s_1^\alpha - h^\alpha . a^{\alpha,\top}) h^\alpha - \sum_{\alpha \neq \gamma} h^\gamma (a^\alpha . h^\gamma) \tag{54}$$

$$\tau \frac{d}{dt} b^\alpha = (s_2^{\alpha,proj} - h^\alpha . b^{\alpha,\top}) b^\alpha - \sum_{\alpha \neq \gamma} h^\gamma (b^\alpha . h^\gamma). \tag{55}$$

Similar to the case in single-task learning, the summations $\sum_{\alpha \neq \gamma}$ represent inter-mode interactions between different layers. The other terms represent the cooperation of the different connectivity modes $h^\alpha, a^\alpha, b^\alpha$ to drive each other to larger magnitudes and similar directions to reflect the strength of the singular values s_1^α and $s_{2,proj}^\alpha$ in the shared weight space. We discuss how the ratio of the singular values s_1^α and $s_{2,proj}^\alpha$ defines the learning dynamics of the connectivity modes and hence how the different tasks are influenced by differences in magnitudes. Projecting everything along the r^α direction ($h^\alpha . r^\alpha = h, a^\alpha . r^\alpha = a, b^\alpha . r^\alpha = b, s_1^\alpha . r^\alpha = s_1$ and $s_{2,proj}^\alpha . r^\alpha = s_{2,proj}$) turns the equations in (55) to the coupled scalar equations,

$$\tau \frac{dh}{dt} = [a(s_1 - ah) + b(s_{2,proj} - bh)] \tag{56}$$

$$\tau \frac{da}{dt} = h(s_1 - ah) \tag{57}$$

$$\tau \frac{db}{dt} = h(s_1^{proj} - bh). \tag{58}$$

For similar tasks, the γ factor occurring in (41) corresponds to a scalar factor. As such we can write $h = Kb(t)$ and $a = Lb(t)$, where L and K are scalar scaling factors which depend on the ratio s_{proj}/s_1. Inserting $h = Kb(t)$ and $a = Lb(t)$ into (56) yields,

$$\frac{db}{b\left(s_1(L + \frac{s_{proj}}{s_1}) - Kb^2(L^2+1)\right)} = \frac{1}{\tau K} dt; \tag{59}$$

or, setting, $K' = K(L^2+1), s' = s_1(L + \frac{s_{proj}}{s_1})$,

$$\frac{db}{b(s' - K'b^2)} = \frac{1}{\tau K} dt. \tag{60}$$

Substituting $u = b^2$ and using partial fractions: $\frac{1}{u(s-uK)} = \frac{1}{su} + \frac{K'}{s(s-uK)}$ yields,

$$\frac{1}{2}\left(\int_{u_0}^{u_f} \frac{du}{s'u} + \int_{u_0}^{u_f} \frac{K' du}{s'(s' - K'u)}\right) = \frac{1}{\tau K'} \int_0^t dt, \tag{61}$$

or after substituting $p = s' - uK'$,

$$\frac{1}{2}\left(\frac{1}{s_{proj}} \ln |u|_{u_0}^{u_f} - \int_{p_0}^{p_f} \frac{dp}{s_{proj} p}\right) = \frac{t}{\tau K} \tag{62}$$

and finally,

$$\frac{K'\tau}{2} \frac{1}{s_{proj}} \left(\ln |\frac{b_f^2}{b_0^2}| + \ln \frac{|(s - K'b_0^2)|}{|(s - K'b_f^2)|}\right) = t. \tag{63}$$

$$\frac{K'\tau}{2s'} \ln \frac{|b_f^2(s' - K'b_0^2)|}{|b_0^2((s' - K'b_f^2)|} = t \tag{64}$$

Inverting this to an expression in terms of b and going back to K and s_{proj} one obtains,

$$b(t) = \sqrt{\frac{e^{\frac{2(s_1 L + s_{proj})t}{\tau K}}(s_1 L + s_{proj})}{\frac{(s_1 L + s_{proj})}{b_0^2} + e^{\frac{2(s_1 L + s_{proj})t}{\tau K}} K(L^2+1) - K(L^2+1)}} \tag{65}$$

$$h(t) = Kb(t) \qquad a(t) = Lb(t), \tag{66}$$

where, K and L are scaling constants,

$$K = \frac{a_K}{\frac{s_{proj}}{s_1} + b_K} \qquad L = \frac{a_L}{\frac{s_{proj}}{s_1} + b_L} \approx \frac{s_1}{s_{proj}}, \tag{67}$$

where the last approximation is a result of noticing that $a_L = 1.006 \approx 1$ and $b_L = 0.006 \approx 0$.

Imbalance in magnitude: $s_1/s_{2,proj} << 1$. When the singular values associated to the same mode are different, in such a way that it is much stronger for task 2 than task 1, the solution in equation (66) can be written as,

$$b_{s_{proj}>s_1}(t) = \sqrt{\frac{e^{\frac{2s_{2,proj}t}{\tau K}} s_{2,proj}}{\frac{s_{2,proj}}{b_0^2} + e^{\frac{2s_{2,proj}t}{\tau K}} K - K}} \tag{68}$$

$$h_{s_{proj}>s_1}(t) = Kb_{s_{proj}>s_1}(t) \qquad a_{s_{proj}>s_1}(t) = Lb_{s_{proj}>s_1}(t) \tag{69}$$

Hence, in this case, the magnitude of the shared connectivity mode of the multi-task system are determined fully by the second task and $s_{2,proj}$. Furthermore, for $K \to 1$, this correponds to the solution derived in Saxe et al. (2014; 2019) for single task learning with $h \approx b$. In this regime, the hidden layer and task-specific layer corresponding to the "leading task" act as if they were in a single-task network. The smaller the relative difference between $s_{2,proj}$ and s_1, the bigger K and hence the difference with the single-task behaviour. Interestingly, $a(t)$ converges at the same speed as h and b, which can be explained by the rise of h that influences a. As can be seen from the $s_1/s_{2,proj}$ factor appearing in $a(t)$, a stays relatively small for big values of $s_{2,proj}$ but the change rate is similar or equal to that of h. Of course, a similar solution holds for the case where $s_1/s_{2,proj} > 1$ by interchanging the roles of a and b.

Similar magnitude singular values $s_1 \approx s_{2,proj}$. When a mode is equally strong for both tasks, i.e. $s_1 = s_{2,proj}$ the solution in (66) can be written as,

$$b_{\sim}(t) = \sqrt{\frac{e^{\frac{4s_{2,proj}t}{\tau K}} s_{2,proj}}{\frac{s_{2,proj}}{b_0^2} + e^{\frac{4s_{2,proj}t}{\tau K}} K - K}} \tag{70}$$

$$h_{\sim}(t) = Kb(t) \qquad a_{\sim}(t) = b(t), \tag{71}$$

In this case, the perceived strength is doubled for both tasks and $L = 1$ so both modes evolve in the same way.

B.6 Learning speed compared to Single Task Learning

By inverting the solution in equation (66), we find that the time to learn modes of strengths s_{proj} in the regime when for $s_{proj} \geq s_1$ is (and therefore also s_1 since a converges at same speed in MTL),

$$t = \frac{K\tau}{2(s_1 L + s_{proj})} \ln \frac{b_f^2\left((s_1 L + s_{proj}) - (K(L^2+1))b_0^2\right)}{b_0^2\left((s_1 L + s_{proj}) - (K(L^2+1))b_f^2\right)} \tag{72}$$

When the difference in strength is significant, $K \approx 1$ and $L << 1$, such that the time corresponds to

$$t_{s_{proj}>\mathbf{s}_1} = \frac{K\tau}{2s_{proj}} \ln \frac{|b_f^2(s_{proj} - Kb_0^2)|}{|b_0^2((s_{proj} - Kb_f^2)|} \tag{73}$$

which for $K = 1$ is equal to the time needed to learn a mode of s_{proj} in a single task network, a shown in (Saxe et al., 2019). However, if $K > 1$ the time needed to learn that same mode is longer in MTL than in STL (but keep in mind we are in fact learning two tasks and two modes). In contrast, in the regime $s_1 \approx s_{proj}$ we learn that the speed is equal to

$$t_{s_{proj}\sim s_1} = \frac{K\tau}{4s_{proj}} \ln \frac{|b_f^2(s_{proj} - Kb_0^2)|}{|b_0^2((s_{proj} - Kb_f^2)|} \tag{74}$$

so for $K \approx 1$, this yields a speed-up of a factor of two as compared to single-task learning. Furthermore, note that in both regimes, the learning speed is determined by the magnitude of the strongest mode.

B.7 Effect of depth

As shown in Saxe et al. (2014; 2019), the main qualitative difference in learning dynamics for deeper linear networks is observed between the case of a shallow network (only one layer and no hidden layer) and deeper networks. The extra layer impacts the learning speed but keeps the stage-like transitions and formatting of internal representations (Saxe et al., 2019). In principle, there is no reason why this should be different in the case of multiple tasks. We confirm this in Appendix D, where we present results on deeper networks.

C Multi-Task Riccati Equations

In this section, we describe another way of solving the MTL gradient flow equations corresponding to the Riccati formalism discussed in the main paper. This formalism allows solutions for a broader family of initial conditions.

Defining,

$$\mathbf{Q}_{MTL} = \begin{pmatrix} \mathbf{W}_S^\top \\ \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \qquad \mathbf{R}_{MTL} = \begin{pmatrix} 0 & \mathbf{\Sigma}_{1o}^\top & \mathbf{\Sigma}_{2o}^\top \\ \mathbf{\Sigma}_{1o} & 0 & 0 \\ \mathbf{\Sigma}_{2o} & 0 & 0 \end{pmatrix}, \tag{75}$$

allows for writing the gradient flow equations as Riccati equations and, under some assumptions, obtaining analytical expressions of the learning dynamics of the task-specific network functions, $\mathbf{W}_i\mathbf{W}_S(t)$ and task-specific training losses $\mathcal{L}_i$ and the representational similarity matrices (RSM).

C.1 Assumptions & Discussion

The derivations discussed in the main paper, which are building on Braun et al. (2022), depend on the following assumptions:

- **Assumption A1.** The inputs are *whitened*, i.e., $\mathbf{\Sigma}_{11} = \mathbf{I}$.
- **Assumption A2.** The weights are *zero-balanced* at initialization: $\mathbf{W}_S(0)\mathbf{W}_S(0)^\top = \mathbf{W}_1(0)^\top\mathbf{W}_1(0) + \mathbf{W}_2(0)^\top\mathbf{W}_2(0)$. **A2bis.** The weights are $\mathbf{\Gamma}$-*balanced* at initialization: $\mathbf{W}_S(0)\mathbf{W}_S(0)^\top - \mathbf{W}_1(0)^\top\mathbf{W}_1(0) - \mathbf{W}_2(0)^\top\mathbf{W}_2(0) = \mathbf{\Gamma}$.
- **Assumption A3.** *No bottleneck:* $\forall$ i $\in$ {1,2}: Rank($\mathbf{\Sigma}_{io}$)=min(N_1, N_{io}), ensuring the shared space can hold all the task-relevant input-output modes for both tasks $N_S \geq \min(N_1, N_{1o} + N_{2o})$.

These are not different from the assumptions in STL literature (apart from Assumption A2, which is extended to multiple tasks). A detailed discussion about these assumptions and possible relaxations is presented in Dominé et al. (2025); Tarmoun et al. (2021). Resulting insights indicate that relaxing them mainly leads to

phenomena associated with transitions between the lazy and rich regimes, a well-known property of neural network training that is not specific to MTL. In particular, recent work Dominé et al. (2025), has extended the Riccati framework to more general initialization schemes. In particular, the zero-balanced initialization assumption has been generalized to a Γ-balanced setting, enabling a smooth interpolation between the rich and lazy regimes and allowing us to analyze how network architecture influences the transition between them. Accordingly, we extend the solution of Theorem 3.2 to the Γ-balanced regime in Appendix C.6, and to bottlenecked networks in Appendix C.7.

C.2 Derivation of the MTL Riccati Equations

Lemma C.1. *Under assumptions A1, the continuous-time dynamics of the matrix $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top$ from an initial state $\mathbf{Q}_{MTL}(0)$ determined by an initialization $\mathbf{W}_S(0), \mathbf{W}_1(0), \mathbf{W}_2(0)$,*

$$\tau \frac{d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)}{dt} = \mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top + \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} - (\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2 \tag{76}$$

$$- \mathbf{Q}_{MTL}\mathbf{\Gamma}(t)\mathbf{Q}_{MTL}^\top, \tag{77}$$

where $\mathbf{\Gamma}$ is a constant matrix describing the difference in covariance between the shared and task-specific weights.

Proof. First note that,

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top = \begin{pmatrix} \mathbf{W}_S^\top\mathbf{W}_S & \mathbf{W}_S^\top\mathbf{W}_1^\top & \mathbf{W}_S^\top\mathbf{W}_2^\top \\ \mathbf{W}_1\mathbf{W}_S & \mathbf{W}_1\mathbf{W}_1^\top & \mathbf{W}_1\mathbf{W}_2^\top \\ \mathbf{W}_2\mathbf{W}_S & \mathbf{W}_2\mathbf{W}_1^\top & \mathbf{W}_2\mathbf{W}_2^\top \end{pmatrix}, \tag{78}$$

such that,

$$\tau \frac{d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)}{dt} = \tau \begin{pmatrix} \frac{d\mathbf{W}_S^\top}{dt}\mathbf{W}_S + \mathbf{W}_S^\top\frac{d\mathbf{W}_S}{dt} & \frac{d\mathbf{W}_S^\top}{dt}\mathbf{W}_1^\top + \mathbf{W}_S^\top\frac{d\mathbf{W}_1^\top}{dt} & \frac{d\mathbf{W}_S^\top}{dt}\mathbf{W}_2^\top + \mathbf{W}_S^\top\frac{d\mathbf{W}_2^\top}{dt} \\ \frac{d\mathbf{W}_1}{dt}\mathbf{W}_S + \mathbf{W}_1\frac{d\mathbf{W}_S}{dt} & \frac{d\mathbf{W}_1}{dt}\mathbf{W}_1^\top + \mathbf{W}_1\frac{d\mathbf{W}_1^\top}{dt} & \frac{d\mathbf{W}_1}{dt}\mathbf{W}_2^\top + \mathbf{W}_1\frac{d\mathbf{W}_2^\top}{dt} \\ \frac{d\mathbf{W}_2}{dt}\mathbf{W}_S + \mathbf{W}_2\frac{d\mathbf{W}_S}{dt} & \frac{d\mathbf{W}_2}{dt}\mathbf{W}_1^\top + \mathbf{W}_2\frac{d\mathbf{W}_1^\top}{dt} & \frac{d\mathbf{W}_2}{dt}\mathbf{W}_2^\top + \mathbf{W}_2\frac{d\mathbf{W}_2^\top}{dt} \end{pmatrix} \tag{79}$$

$$\equiv \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix}. \tag{80}$$

Assuming Whitened Inputs (Assumption A1) allows us to use the equations (31). Then, one obtains the explicit expression of the components A-I in $d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)/dt$,

$$\begin{aligned}
A &= (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2^\top\mathbf{W}_2)\mathbf{W}_S + \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_2^\top\mathbf{\Sigma}_2) \\
&\quad - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \\
B &= (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top + \mathbf{W}_S^\top\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top \\
&\quad - \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top \\
C &= (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top + \mathbf{W}_S^\top\mathbf{W}_S\mathbf{\Sigma}_2^\top - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top \\
&\quad - \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top \\
D &= \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{\Sigma}_1^\top + \mathbf{W}_2^\top\mathbf{\Sigma}_2^\top) - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S \\
&\quad - \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \\
E &= \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top \\
F &= \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_2^\top + \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_2^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top \\
G &= \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_2(\mathbf{W}_1^\top\Sigma_1^\top + \mathbf{W}_2^\top\mathbf{\Sigma}_2^\top) - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S \\
&\quad - \mathbf{W}_2(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \\
H &= \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_2\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top \\
I &= \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_2\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top
\end{aligned}$$

Essentially, we have to show that the components of the right-hand side of equation (77),

$$\mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top + \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} - (\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2 - \mathbf{Q}_{MTL}\mathbf{\Gamma}_0\mathbf{Q}_{MTL} \equiv \begin{pmatrix} A' & B' & C' \\ D' & E' & F \\ G' & H' & I' \end{pmatrix}, \quad (81)$$

is equal to the left-hand side. In particular, we show that components (A'-I') are equal to components (A-I). To do so, note that

$$\mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top = \begin{pmatrix} \mathbf{\Sigma}_1^\top\mathbf{W}_1\mathbf{W}_S + \mathbf{\Sigma}_2^\top\mathbf{W}_2\mathbf{W}_S & \mathbf{\Sigma}_1^\top\mathbf{W}_1\mathbf{W}_1^\top + \mathbf{\Sigma}_2^\top\mathbf{W}_2\mathbf{W}_1^\top & \mathbf{\Sigma}_1^\top\mathbf{W}_1\mathbf{W}_1^\top + \mathbf{\Sigma}_2^\top\mathbf{W}_2\mathbf{W}_2^\top \\ \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_S & \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_1^\top & \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_2^\top \\ \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_S & \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_1^\top & \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_2^\top \end{pmatrix} \quad (82)$$

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} = \begin{pmatrix} \mathbf{W}_S^\top\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_S^\top\mathbf{W}_2^\top\mathbf{\Sigma}_2 & \mathbf{W}_S^\top\mathbf{W}_S\Sigma_1^\top & \mathbf{W}_S^\top\mathbf{W}_S\Sigma_2^\top \\ \mathbf{W}_1\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_1\mathbf{W}_2^\top\mathbf{\Sigma}_2 & \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_1^\top & \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_2^\top \\ \mathbf{W}_2\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_2\mathbf{W}_2^\top\mathbf{\Sigma}_2 & \mathbf{W}_2\mathbf{W}_S\mathbf{\Sigma}_1^\top & \mathbf{W}_2\mathbf{W}_S\mathbf{\Sigma}_2^\top \end{pmatrix} \quad (83)$$

$$(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2 \equiv \begin{pmatrix} A'' & B'' & C'' \\ D'' & E'' & F'' \\ G'' & H'' & I'' \end{pmatrix} \quad (84)$$

with (85)

$$A'' = \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \quad (86)$$

$$B'' = \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^{\mathbf{1}^\top + \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top} \quad (87)$$

$$C'' = \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^{\mathbf{2}^\top + \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top} \quad (88)$$

$$D'' = \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \quad (89)$$

$$E'' = \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top \quad (90)$$

$$F'' = \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top + \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top \quad (91)$$

$$G'' = \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top + \mathbf{W}_2\mathbf{W}_1^\top\mathbf{W}_1\mathbf{W}_2^\top + \mathbf{W}_2\mathbf{W}_2^\top\mathbf{W}_2\mathbf{W}_2^\top \quad (92)$$

$$H'' = \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_2\mathbf{W}_1^\top\mathbf{W}_1\mathbf{W}_1^\top + \mathbf{W}_2\mathbf{W}_2^\top\mathbf{W}_2\mathbf{W}_1^\top \quad (93)$$

$$I'' = \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S^\top + \mathbf{W}_2\mathbf{W}_1^\top\mathbf{W}_1\mathbf{W}_S + \mathbf{W}_2\mathbf{W}_2^\top\mathbf{W}_2\mathbf{W}_S \quad (94)$$

Such that,

$$A' = (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2^\top\mathbf{W}_2)\mathbf{W}_S + \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_2^\top\mathbf{\Sigma}_2) \quad (95)$$

$$- \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_S^\top\mathbf{W}_2)\mathbf{W}_S \quad (96)$$

$$B' = (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2\mathbf{W}_2)\mathbf{W}_1^\top + \mathbf{W}_S^\top\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_S^\top\mathbf{W}_2)\mathbf{W}_1^\top \quad (97)$$

$$- \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top \quad (98)$$

$$C' = (\mathbf{\Sigma}_1^\top\mathbf{W}_1 + \mathbf{\Sigma}_2\mathbf{W}_2)\mathbf{W}_2^\top + \mathbf{W}_S^\top\mathbf{W}_S\mathbf{\Sigma}_2^\top - \mathbf{W}_S^\top(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_S^\top\mathbf{W}_2)\mathbf{W}_2^\top \quad (99)$$

$$- \mathbf{W}_S^\top\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top \quad (100)$$

$$D' = \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{\Sigma}_1^\top + \mathbf{W}_2^\top\mathbf{\Sigma}_2^\top) - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S \quad (101)$$

$$- \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_S^\top\mathbf{W}_2)\mathbf{W}_S \quad (102)$$

$$E' = \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top - \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top \quad (103)$$

$$F' = \mathbf{\Sigma}_1\mathbf{W}_S^\top\mathbf{W}_2^\top + \mathbf{W}_1\mathbf{W}_S\mathbf{\Sigma}_2^\top - \mathbf{W}_1\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top - \mathbf{W}_1(\mathbf{W}_1^\top\mathbf{W}_1 - \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top \quad (104)$$

$$G' = \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_S + \mathbf{W}_2(\mathbf{W}_1^\top\mathbf{\Sigma}_1 + \mathbf{W}_2^\top\mathbf{\Sigma}_2) - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_S \quad (105)$$

$$- \mathbf{W}_2(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_S \quad (106)$$

$$H' = \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_1^\top + \mathbf{W}_2\mathbf{W}_S\mathbf{\Sigma}_1^\top - \mathbf{W}_S\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_1^\top - \mathbf{W}_2(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_1^\top \quad (107)$$

$$I' = \mathbf{\Sigma}_2\mathbf{W}_S^\top\mathbf{W}_2^\top + \mathbf{W}_S\mathbf{W}_S\mathbf{\Sigma}_2^\top - \mathbf{W}_2\mathbf{W}_S\mathbf{W}_S^\top\mathbf{W}_2^\top - \mathbf{W}_2(\mathbf{W}_1^\top\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{W}_2)\mathbf{W}_2^\top \quad (108)$$

For components B, C, D, G of equation (77) the equality to B', C', D', G' of the right hand side is trivial without any further assumption. In contrast, the other components (A, E, F, H, I) differ from their counterparts (A', E', F', H', I') by the difference in covariance between the shared weights and task-specific weights, which, under assumption, we can write as

$$\mathbf{W}_S\mathbf{W}_S^\top - \mathbf{W}_1^\top\mathbf{W}_1 - \mathbf{W}_2^\top\mathbf{W}_2 \equiv \mathbf{\Gamma}(t). \tag{109}$$

Using equation (109) in the remaining components (A', E', F', H', I') proves the Lemma.

□

Lemma C.2. *Assuming whitened inputs (assumption A1) and balanced weights at initialization (assumption A2bis) with a charge equal to $\mathbf{\Gamma}_0$, the gradient flow equations of $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}$ can be written as,*

$$\tau\frac{d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)}{dt} = \mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top + \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} - (\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2 \tag{110}$$

$$- \mathbf{Q}_{MTL}^\top\mathbf{\Gamma}_0\mathbf{Q}_{MTL}, \tag{111}$$

Proof. As shown in Dominé et al. (2025), $\mathbf{\Gamma} \in \mathbb{R}^{N_s \times N_s}$ is constant, i.e., so the imbalance is determined by the initial imbalance $\mathbf{\Gamma}_0$. Using this in the proof of Lemma C.1 proves equation (111). □

As shown in Dominé et al. (2025), the charge $\mathbf{\Gamma}_0$ controls the behavior between the Lazy and Rich Regime. In this work, we focus on the inter-task dynamics of zero-balanced weights (i.e., $\mathbf{\Gamma}_0 = 0$) which constraints the model to the Rich Regime.

Lemma C.3. *Assuming whitened inputs (assumption A1) and zero-balanced weights at initialization (assumption A2bis), the gradient flow equations of $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}$ can be written as,*

$$\tau\frac{d(\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)}{dt} = \mathbf{R}_{MTL}\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top + \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top\mathbf{R}_{MTL} - (\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top)^2. \tag{112}$$

Proof. Replacing $\mathbf{\Gamma}_0$ with a matrix filled with zero's in equation (111) proves the statement. □

C.3 Solution to the MTL Riccati Equations: Zero-Balanced Weights

As is the case in Fukumizu (1998); Braun et al. (2022), this work primarily focuses on the setting where the shared and task-specific weights are zero-balanced ($\mathbf{\Gamma}_0 = 0_{N_S \times N_S}$).

Lemma C.4. *Under assumption A1 (whitened inputs), assumption A2bis (zero-balanced weights) and assumption A3 (full-rank), the MTL Riccati equation (77) has a unique solution $\forall t > 0$ for the case where $N_{1o} + N_{2o} = N_1$, given by*

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t) \tag{113}$$

$$= e^{\mathbf{R}_{MTL}\frac{t}{\tau}}\mathbf{Q}_{MTL}(0)\left[\mathcal{I} + \frac{1}{2}\mathbf{Q}_{MTL}(0)^\top(e^{\mathbf{R}_{MTL}\frac{t}{\tau}}\mathbf{R}_{MTL}^{-1}e^{\mathbf{R}_{MTL}\frac{t}{\tau}} - \mathbf{R}_{MTL}^{-1})\mathbf{Q}_{MTL}(0)\right]^{-1} \tag{114}$$

$$\times\, \mathbf{Q}_{MTL}(0)^{\mathbf{R}_{MTL}\frac{t}{\tau}}, \tag{115}$$

where $Q(0)$ is assumed to be full rank and corresponds to the initial state.

Proof. As shown in Fukumizu (1998), equation (112) has a unique solution under assumptions A1-A2-A3. This can be shown explicitly by inserting equation (115) into (112). □

Note that assumption IV can be relaxed for the theorem, where the approach in Braun et al. (2022) is used to rewrite the solution (115) in a more interpretable way. Namely, defining,

$$\xi = \begin{pmatrix}\mathbf{\Sigma}_{1o}\\ \mathbf{\Sigma}_{2o}\end{pmatrix} \qquad \mathbf{W}_{MTL}^0 = \begin{pmatrix}\mathbf{W}_1(0)\mathbf{W}_S(0)\\ \mathbf{W}_2(0)\mathbf{W}_S(0)\end{pmatrix}, \tag{116}$$

such that $\xi, \mathbf{W}^0_{MTL} \in \mathbb{R}^{N_{1o}+N_{2o}\times N_1}$, allows us to consider the compact singular value decomposition of the initial network and the input-output correlation matrix of the shared-task space,

$$\xi = \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^\top \qquad \mathbf{W}^0_{MTL} = \mathbf{U}_0 \mathbf{S}_0 \mathbf{V}_0. \tag{117}$$

Then $\mathbf{S}_\xi, \mathbf{S}_0 \in \mathbb{R}^{m\times m}$ with $m = min(N_{1o}+N_{2o}, N_1)$ represents the combined strength of the correlations contained in the two input-output matrices $\mathbf{\Sigma}_{1o}, \mathbf{\Sigma}_{2o}$, they are not simply the singular values of the individual matrices but rather reflect their joint contribution. Furthermore, $\mathbf{U}_\xi \in \mathbb{R}^{N_{1o}+N_{2o}\times m}$ and $\mathbf{V}_\xi^\top \in \mathbb{R}^{m\times N_1}$. Similar to the single-task case discussed in Braun et al. (2022), if $N_1 \neq N_{1o}+N_{2o}$, then, in general, $\mathbf{U}_\xi$ and $\mathbf{V}_\xi$ are not square and orthonormal. Namely, if $N_1 < N_{1o}+N_{2o}$, then $m = N_1$ such that $\mathbf{U}_\xi^\top \mathbf{U}_\xi = \mathbf{V}_\xi^\top \mathbf{V}_\xi = \mathbf{V}_\xi \mathbf{V}_\xi^\top = \mathcal{I} \in \mathbb{R}^{N_1\times N_1}$ while $\mathbf{U}_\xi \mathbf{U}_\xi^\top \neq \mathbf{I}$ in $\mathbb{R}^{(N_{1o}+N_{2o})\times(N_{1o}+N_{2o})}$. In that case, we can define $\mathbf{U}_\xi^\perp \in \mathbb{R}^{N_{1o}+N_{2o}\times N_{1o}+N_{2o}-N_1}$ to denote the matrix that contains orthogonal column vectors such that concatenation $[U_\xi U_\xi^\perp] \in \mathbb{R}^{N_{1o}+N_{2o}, N_{1o}+N_{2o}}$ is orthonormal. In that case, $V_\xi^\perp \in \mathbb{R}^{N_1\times N_{1o}+N_{2o}-N_i}$ denotes a matrix of zeros.

Conversely, for the situation: $N_1 > N_{1o}+N_{2o}$, in that case, $m = N_{1o}+N_{2o}$ such that $U_\xi U_\xi^\top = U_\xi^\top U_\xi = \mathbf{V}_\xi^\top V = \mathbf{I} \in \mathbb{R}^{(N_{1o}+N_{2o})\times(N_{1o}+N_{2o})}$ but $\mathbf{V}_\xi \mathbf{V}_\xi^\top \neq I \in \mathbb{R}^{N_1\times N_1}$. Defining $V_\xi^\perp \in \mathbb{R}^{N_1\times(N_1-N_{1o}+N_{2o})}$ such that concat $[\mathbf{V}_\xi \mathbf{V}_\xi^\perp] \in \mathbb{R}^{N_1\times N_1}$ is orthonormal and $U_{\xi^\perp} \in \mathbb{R}^{N_{1o}+N_{2o}\times N_{1o}+N_{2o}-N_1}$ denotes a matrix of zeros.
Finally introducing $\mathbf{A}_\xi$ and $\mathbf{B}_\xi$,

$$\mathbf{A}_\xi = \mathbf{U}_0^\top \mathbf{U}_\xi + \mathbf{V}_0^\top \mathbf{V}_\xi \qquad \mathbf{B}_\xi = \mathbf{U}_0^\top \mathbf{U}_\xi - \mathbf{V}_0^\top \mathbf{V}_\xi, \tag{118}$$

allows an interpretable solution for $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$, as formulated in the next Theorem 3.2.

Theorem C.5. *Under assumption A1 (whitened inputs), assumption A2(zero-balanced weights), assumption A3 (full-rank) and non-singular $\mathbf{A}_\xi$, the MTL Riccati equation (77) has a unique solution $\forall t > 0$, given by,*

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t) \tag{119}$$

$$= \mathbf{Z}_{MTL}\Bigg[4\, e^{-\mathbf{S}_\xi \frac{t}{\tau}}\, \mathbf{A}_\xi^{-1}\, \mathbf{S}_\xi^{-1} \left(\mathbf{A}_\xi^T\right)^{-1} e^{-\mathbf{S}_\xi \frac{t}{\tau}} \;+\; \left(\mathbf{I} - e^{-2\mathbf{S}_\xi \frac{t}{\tau}}\right) \mathbf{S}_\xi^{-1} \tag{120}$$

$$- e^{-\mathbf{S}_\xi \frac{t}{\tau}}\, \mathbf{A}_\xi^{-1}\, \mathbf{B} \left(e^{-2\mathbf{S}_\xi \frac{t}{\tau}} - \mathbf{I}\right) \mathbf{S}_\xi^{-1} \mathbf{B}^\top \left(\mathbf{A}_\xi^T\right)^{-1} e^{-\mathbf{S}_\xi \frac{t}{\tau}} \tag{121}$$

$$+ 4\,\frac{t}{\tau}\, e^{-\mathbf{S}_\xi \frac{t}{\tau}}\, \mathbf{A}_\xi^{-1} \left(\mathbf{V}_\xi^\top\, \mathbf{V}_\xi^\perp (\mathbf{V}_\xi^\perp)^\top\, \mathbf{V}_\xi \;+\; \mathbf{U}_\xi^\top\, \mathbf{U}_\xi^\perp (\mathbf{U}_\xi^\perp)^\top\, \mathbf{U}_\xi\right) \left(\mathbf{A}_\xi^T\right)^{-1} e^{-\mathbf{S}_\xi \frac{t}{\tau}}\Bigg]^{-1} \mathbf{Z}_{MTL}^\top, \tag{122}$$

$$\mathbf{Z}_{MTL} \;=\; \begin{pmatrix} \mathbf{V}_\xi \left(\mathbf{I} \;-\; e^{-\mathbf{S}_\xi \frac{t}{\tau}}\, \mathbf{B}_\xi^\top\, (\mathbf{A}_\xi^T)^{-1}\, e^{-\mathbf{S}_\xi \frac{t}{\tau}}\right) \;+\; 2\,\mathbf{V}_\xi^\perp (\mathbf{V}_\xi^\perp)^\top\, \mathbf{V}_\xi\, (\mathbf{A}_\xi^T)^{-1}\, e^{-\mathbf{S}_\xi \frac{t}{\tau}} \\ \mathbf{U}_\xi \left(\mathbf{I} \;+\; e^{-\mathbf{S}_\xi \frac{t}{\tau}}\, \mathbf{B}_\xi^\top (\mathbf{A}_\xi^\top)^{-1}\, e^{-\mathbf{S}_\xi \frac{t}{\tau}}\right) \;+\; 2\,\mathbf{U}_\xi^\perp (\mathbf{U}_\xi^\perp)^\top\, \mathbf{U}_\xi\, (\mathbf{A}_\xi^\top)^{-1}\, e^{-\mathbf{S}_\xi \frac{t}{\tau}} \end{pmatrix}. \tag{123}$$

Proof. Essentially, the proof consists of using the singular value decomposition of $\mathbf{R}_{MTL}$ to obtain explicit expressions for $\mathbf{R}_{MTL}^{-1}$ and $e^{\mathbf{R}_{MTL}}$ in terms of $\mathbf{S}_\xi$, yielding a more interpretable and stable analytical solution.

To do so, rewrite $\mathbf{R}_{MTL}$ as a 2×2 block-matrix. By completing the basis of the resulting block-matrix, one can take the same approach as used in Appendix C in Braun et al. (2022) to diagonalize F,

$$\mathbf{R}_{MTL} = \begin{pmatrix} 0 & \mathbf{\Sigma}_{1o}^\top & \mathbf{\Sigma}_{2o}^\top \\ \mathbf{\Sigma}_{1o} & 0 & 0 \\ \mathbf{\Sigma}_{2o} & 0 & 0 \end{pmatrix} \tag{124}$$

$$= \begin{pmatrix} 0 & \xi^\top \\ \xi & 0 \end{pmatrix} \tag{125}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{V}_\xi & \mathbf{V}_\xi & \sqrt{2}\mathbf{V}_\xi^\perp \\ \mathbf{U}_\xi & -\mathbf{U}_\xi & \sqrt{2}\mathbf{U}_\xi^\perp \end{pmatrix} \begin{pmatrix} \mathbf{S}_\xi & 0 & 0 \\ 0 & -\mathbf{S}_\xi & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{V}_\xi & \mathbf{V}_\xi & \sqrt{2}\mathbf{V}_\xi^\perp \\ \mathbf{U}_\xi & -\mathbf{U}_\xi & \sqrt{2}\mathbf{U}_\xi^\perp \end{pmatrix}^\top \tag{126}$$

$$= \mathbf{P}_{MTL}\Gamma_{MTL}\mathbf{P}_{MTL}^\top \tag{127}$$

Then, $\mathbf{P}_{MTL}\mathbf{P}_{MTL}^\top = \mathbf{I}$ and therefore $\mathbf{P}_{MTL}^\top = \mathbf{P}_{MTL}^{-1}$, enabling the computation of $\mathbf{R}_{MTL}^{-1}$ and $e^{R\frac{t}{\tau}}$ in equation (115). The remaining part of the proof is analogous to the proof presented in Appendix C2 of Braun et al. (2022) without any further assumptions. □

C.4 Proof of the Convergence Behaviour

Theorem C.6. *Under the assumptions of Theorem 3.2,* $\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t)$ *converges to a steady state given by,*

$$\mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t_{steady}) = \begin{pmatrix} \mathbf{V}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^T & \mathbf{V}_\xi \mathbf{S}_\xi \mathbf{U}_\xi^T \\ \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^T & \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{U}_\xi^T \end{pmatrix}. \tag{128}$$

Proof. For $t \to \infty$ the negative exponentials appearing in equation (123) converge to zero $\lim_{t\to\infty} e^{-\mathbf{S}_\xi \frac{t}{\tau}} = 0$. Using this in equation (123) yields,

$$\lim_{t\to\infty} \mathbf{Q}_{MTL}\mathbf{Q}_{MTL}^\top(t) = \begin{pmatrix} \mathbf{V}_\xi \\ \mathbf{U}_\xi \end{pmatrix} \mathbf{S}_\xi^{-1} \begin{pmatrix} \mathbf{V}_\xi^\top & \mathbf{U}_\xi^\top \end{pmatrix}, \tag{129}$$

which proves the lemma. □

C.5 Extension towards more tasks

In principle, the derivations also hold for more than two tasks ($N_T > 2$). In the case of an arbitrary number of tasks, one replaces $\mathbf{Q}_{MTL}$ and $\mathbf{R}_{MTL}$ to

$$\mathbf{Q}_{MTL} = \begin{pmatrix} \mathbf{W}_S^\top \\ \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_{N_T} \end{pmatrix} \qquad \mathbf{R}_{MTL} = \begin{pmatrix} 0 & \mathbf{\Sigma}_{1o}^\top & \mathbf{\Sigma}_{2o}^\top & \dots & \mathbf{\Sigma}_{N_T o}^\top \\ \mathbf{\Sigma}_{1o} & 0 & 0 & \dots & 0 \\ \mathbf{\Sigma}_{2o} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \mathbf{\Sigma}_{N_T o} & \dots & \dots & \dots & 0 \end{pmatrix}, \tag{130}$$

and change ξ and $\mathbf{W}_{MTL}^0$ accordingly. Since our analysis focuses on pairwise interactions, introducing more tasks simply amounts to applying the same pairwise analysis to each task pair.

C.6 Extension to Γ-balancedness

In Dominé et al. (2025), the Riccati solutions are derived for non-zero balanced initialisations, i.e.,

$$\mathbf{W}_S(0)\mathbf{W}_S(0)^\top - \mathbf{W}_1^\top(0)\mathbf{W}_1(0) - \mathbf{W}_2(0)^\top \mathbf{W}_2(0) = \lambda \mathbf{I}, \tag{131}$$

allowing the MTL network to move beyond the rich-regime (forced by the zero-balancedness).
To this end, we need to rewrite $\mathbf{R}_{MTL}$,

$$\mathbf{Q}_{MTL} = \begin{pmatrix} \mathbf{W}_S^\top \\ \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \qquad \mathbf{R}_{MTL} = \begin{pmatrix} -\frac{\lambda}{2}\mathbf{I}_{N_1} & \mathbf{\Sigma}_{1o}^\top & \mathbf{\Sigma}_{2o}^\top \\ \mathbf{\Sigma}_{1o} & \frac{\lambda}{2}\mathbf{I}_{N_{o1},N_{o1}} & 0 \\ \mathbf{\Sigma}_{2o} & 0 & \frac{\lambda}{2}\mathbf{I}_{N_{o2},N_{o2}} \end{pmatrix}, \tag{132}$$

and consider the following eigendecomposition of $\mathbf{R}_{MTL}$

$$\mathbf{P} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{V}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi \mathbf{G}_\xi) & \mathbf{V}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi \mathbf{G}_\xi) & \sqrt{2}\,\mathbf{V}_{\xi\perp} \\ \mathbf{U}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi \mathbf{G}_\xi) & -\mathbf{U}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi \mathbf{G}_\xi) & \sqrt{2}\,\mathbf{U}_{\xi\perp} \end{pmatrix}, \qquad \mathbf{\Lambda} = \begin{pmatrix} \mathbf{S}_{\xi,\lambda} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S}_{\xi,\lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_\perp \end{pmatrix}.$$

where the diagonal matrices $\mathbf{S}_{\xi,\lambda}$, $\lambda_\perp$, $\mathbf{H}_\xi$, and $\mathbf{G}_\xi$ are defined as

$$\mathbf{S}_{\xi,\lambda} = \sqrt{\mathbf{S}_\xi^2 + \frac{\lambda^2}{4}\mathbf{I}}, \qquad \lambda_\perp = \operatorname{sgn}(N_{o_1} + N_{o_2} - N_i)\,\frac{\lambda}{2}\,\mathbf{I}_{|N_{oi} - N_1|}, \tag{133}$$

$$\mathbf{H}_\xi = \operatorname{sgn}(\lambda)\sqrt{\frac{\mathbf{S}_{\xi,\lambda} - \mathbf{S}_\xi}{\mathbf{S}_{\xi,\lambda} + \mathbf{S}_\xi}}, \qquad \mathbf{G}_\xi = \frac{1}{\sqrt{\mathbf{I} + \mathbf{H}_\xi^2}}. \tag{134}$$

As explained in detail in Dominé (2025), this eigencomposition of $\mathbf{R}_{MTL}$ allows to write the Riccati solution (equation 115 into four quadrants. First it is useful to define the following

$$\mathbf{B} = (\mathbf{W}_1, \mathbf{W}_2)^\top(0)\, \mathbf{U}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi) + \mathbf{W}_S(0)\mathbf{V}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi) \in \mathbb{R}^{N_S \times N_S}, \tag{135}$$

$$\mathbf{C} = (\mathbf{W}_1, \mathbf{W}_2)^\top(0)\, \mathbf{U}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi) - \mathbf{W}_S(0)\mathbf{V}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi) \in \mathbb{R}^{N_S \times N_S}, \tag{136}$$

$$\mathbf{D} = (\mathbf{W}_1, \mathbf{W}_2)^\top(0)\, \mathbf{U}_{\xi\perp} + \mathbf{W}_S(0)\mathbf{V}_{\xi\perp} \in \mathbb{R}^{N_S \times (N_o - N_i)}. \tag{137}$$

And then, following the steps in Dominé (2025), one can proof the following theorem.

Theorem C.7. *Under the assumptions of (A1) whitened inputs, (A2bis) λ-balanced weights, and (A3) no bottleneck, the temporal dynamics of $\mathbf{QQ}^\top$ are*

$$\mathbf{QQ}^\top(t) = \begin{pmatrix} \mathbf{Z}_1(t)\mathbf{A}^{-1}(t)\mathbf{Z}_1^\top(t) & \mathbf{Z}_1(t)\mathbf{A}^{-1}(t)\mathbf{Z}_2^\top(t) \\ \mathbf{Z}_2(t)\mathbf{A}^{-1}(t)\mathbf{Z}_1^\top(t) & \mathbf{Z}_2(t)\mathbf{A}^{-1}(t)\mathbf{Z}_2^\top(t) \end{pmatrix},$$

with the time-dependent variables $\mathbf{Z}_1(t) \in \mathbb{R}^{N_i \times N_S}$, $\mathbf{Z}_2(t) \in \mathbb{R}^{N_o \times N_S}$, and $\mathbf{A}(t) \in \mathbb{R}^{N_S \times N_S}$:

$$\mathbf{Z}_1(t) = \tfrac{1}{2}\mathbf{V}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi)e^{\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}}\mathbf{B}^\top - \tfrac{1}{2}\mathbf{V}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi)e^{-\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}}\mathbf{C}^\top + \mathbf{V}_{\xi\perp}e^{\lambda_\perp\frac{t}{\tau}}\mathbf{D}^\top, \tag{138}$$

$$\mathbf{Z}_2(t) = \tfrac{1}{2}\mathbf{U}_\xi(\mathbf{G}_\xi + \mathbf{H}_\xi)e^{\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}}\mathbf{B}^\top + \tfrac{1}{2}\mathbf{U}_\xi(\mathbf{G}_\xi - \mathbf{H}_\xi)e^{-\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}}\mathbf{C}^\top + \mathbf{U}_{\xi\perp}e^{\lambda_\perp\frac{t}{\tau}}\mathbf{D}^\top, \tag{139}$$

$$\mathbf{A}(t) = \mathbf{I} + \mathbf{B}\left(\frac{e^{2\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}} - \mathbf{I}}{4\mathbf{S}_{\xi,\lambda}}\right)\mathbf{B}^\top - \mathbf{C}\left(\frac{e^{-2\mathbf{S}_{\xi,\lambda}\frac{t}{\tau}} - \mathbf{I}}{4\mathbf{S}_{\xi,\lambda}}\right)\mathbf{C}^\top + \mathbf{D}\left(\frac{e^{\lambda_\perp\frac{t}{\tau}} - \mathbf{I}}{\lambda_\perp}\right)\mathbf{D}^\top. \tag{140}$$

We use this solution to simulate the learning dynamics for different degrees of alignment (Figure 6-7-8). Finally, we note that this notion of balanced initialisations is well-motivated in practice. As demonstrated in Dominé et al. (2025), standard initialisation schemes such as LeCun or He initialisation yield balanced behaviour in the infinite-width limit.

C.6.1 Impact of λ-balancedness on MTL dynamics

To understand how the sign of λ in

$$\mathbf{W}_S\mathbf{W}_S^\top - \mathbf{W}_1^\top\mathbf{W}_1 - \mathbf{W}_2^\top\mathbf{W}_2 = \lambda\mathbf{I} \tag{141}$$

determines how the insights from the main paper adapt, we need a new algorithm to initialise the networks. Especially, the strength of task-specific network functions need to be preserved, yielding per-direction equations,

$$h_i^2 - a_i^2 - b_i^2 = \lambda\mathbf{I} \tag{142}$$

$$h_i a_i = s_{1;i}^2 \tag{143}$$

$$h_i b_i = s_{2,i}^2 \tag{144}$$

yielding,

$$h_i^2 = \frac{\lambda + \sqrt{\lambda^2 + 4(s_{1,i}^2 + s_{2,i}^2)}}{2} \tag{145}$$

$$a_i^2 = \frac{2s_{1,i}^4}{\lambda + \sqrt{\lambda^2 + 4(s_{1,i}^2 + s_{2,i}^2)}} \tag{146}$$

$$b_i^2 = \frac{2s_{2,i}^4}{\lambda + \sqrt{\lambda^2 + 4(s_{1,i}^2 + s_{2,i}^2)}}. \tag{147}$$

Using this algorithm to initialise the weights, we analyze the dependence of relative task importance on λ-balancedness, and investigate how the sign of λ governs this effect. We consider fully aligned tasks, conflicting and orthogonal tasks created by the procedure described in Appendix D.1.

Fully Aligned Tasks Figure 6 shows the training dynamics for $\lambda = 2, 0, -2$ in the case of fully aligned tasks with different scales, trained in a two-layer linear neural network. Notably, the simultaneity of learning in the balanced case ($\lambda = 0$) is preserved across all settings. As expected, non-zero balancedness leads to faster learning. The main qualitative differences appear in the cosine similarities and gradient magnitudes: for $\lambda = -2$, the cosine similarity remains close to zero (but positive), in contrast to the typically higher positive values observed for $\lambda = 0$ and $\lambda = 2$. Nevertheless, gradient magnitudes for both tasks still grow simultaneously across all values of λ.

Fully Orthogonal Tasks Figure 8 presents the training dynamics for $\lambda = 2, 0, -2$ in the case of orthogonal tasks with imbalanced scale, trained in a neural network. For the balanced case ($\lambda = 0$), a clear gap emerges between the two tasks in terms of network functions, gradient magnitudes, and losses. In the lazy regime, where features are learned simultaneously, this gap disappears for $\lambda = -2$, leading to immediate and simultaneous learning of both tasks, although the cosine similarity remains close to zero. In contrast, for $\lambda = 2$, the second task still converges later, but the gap is substantially reduced, while cosine similarities remain near zero across all regimes. As before, negative balancedness leads to simultaneous convergence.

Fully Conflicting tasks Finally we also include experiments on Fully conflicting tasks in Figure 7.

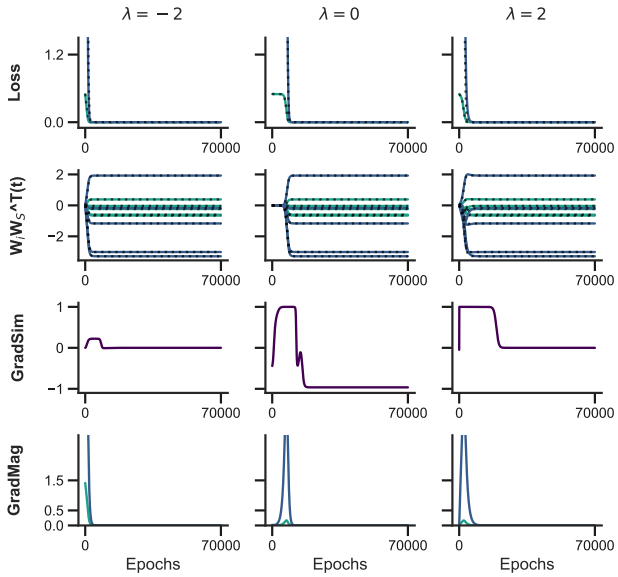

Figure 6: Analytical and numerical simulations of the learning dynamics for $\lambda = 2, 0, -2$ for fully aligned and imbalanced (1,5) tasks trained on a MTL network with $N_1 = 5, N_S = 2; N_{oi} = 2, N = 10, \eta = 0.0002$ for $\lambda = -2, 0, 2$ (from left to right).

C.7 Extension to Bottlenecked Networks

Following Dominé (2025), who extended the single-task analysis to a bottlenecked setting, one can relax the restriction on the hidden dimension in the MTL setting (Assumption A3), in contrast to Braun et al. (2022); Dominé et al. (2025). Essentially one has to replace the definitions in equation 134 to

$$\mathbf{S}_{\xi,\lambda} = \sqrt{\mathbf{S}_\xi^2 + \frac{\lambda^2}{4}\mathbf{I}_{\min(N_1, N_{o1}+N_{o1})}}, \qquad \lambda_\perp = \mathrm{sgn}(N_{o_1} + N_{o_2} - N_i)\,\frac{\lambda}{2}\,\mathbf{I}_{|N_o - N_i|}, \tag{148}$$

$$\mathbf{H}_\xi = \mathrm{sgn}(\lambda)\sqrt{\frac{\mathbf{S}_{\xi,\lambda} - \mathbf{S}_\xi}{\mathbf{S}_{\xi,\lambda} + \mathbf{S}_\xi}}, \qquad \mathbf{G}_\xi = \frac{1}{\sqrt{\mathbf{I}_{\min(\mathbf{N_1}, \mathbf{N_{o1}}+\mathbf{N_{o1}})} + \mathbf{H}_\xi^2}}. \tag{149}$$

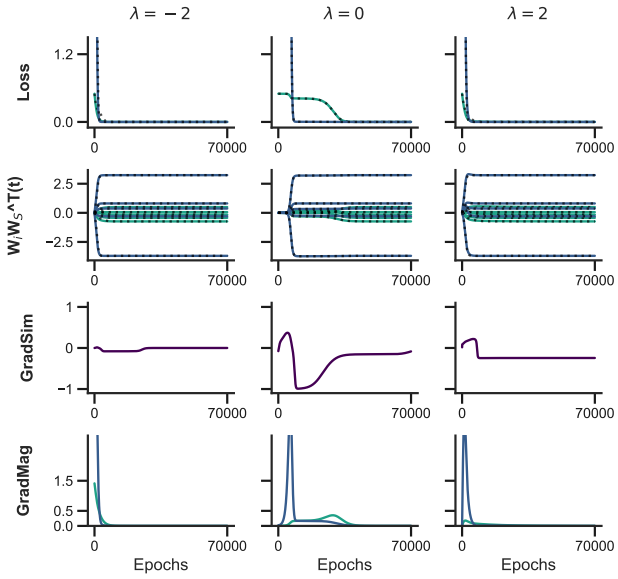

Figure 7: Analytical and numerical simulations of the learning dynamics for $\lambda = 2, 0, -2$ for fully conflicting and imbalanced (1,5) tasks trained on a MTL network with $N_1 = 5, N_S = 2; N_{oi} = 2, N = 10, \eta = 0.0002$ for $\lambda = -2, 0, 2$ (from left to right).

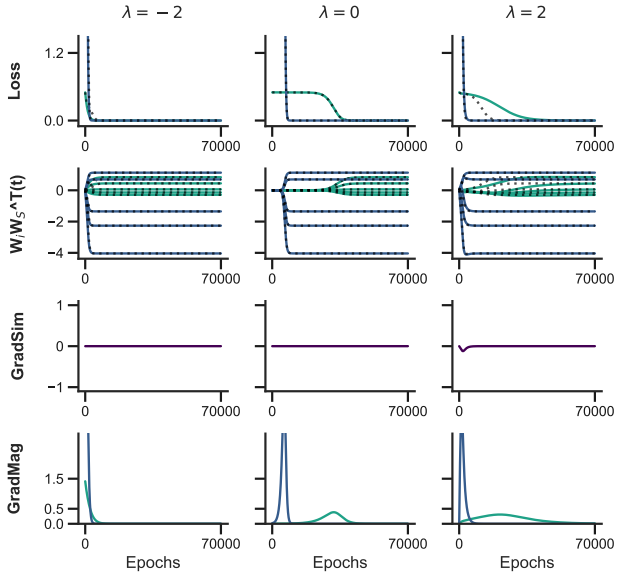

Figure 8: Analytical and numerical simulations of the learning dynamics for $\lambda = -2, 0, 2$ for fully Orthogonal and imbalanced (1,5) tasks trained on a MTL network with $N_1 = 5, N_S = 2; N_{oi} = 2, N = 10, \eta = 0.0002$ for $\lambda = -2, 0, 2$ (from left to right).

We leave further details for the reader.

This relaxation allows us to analyse a broader and more realistic class of architectures. Figures 9 and C.7 show fully aligned and fully orthogonal tasks, constructed by the procedure in Appendix D.1. We study bottlenecked networks ($N_1 = 10$, $N_S = 1$, $N_{o_1} = 2$, $N_{o_2} = 2$) with zero-balanced initialisation.

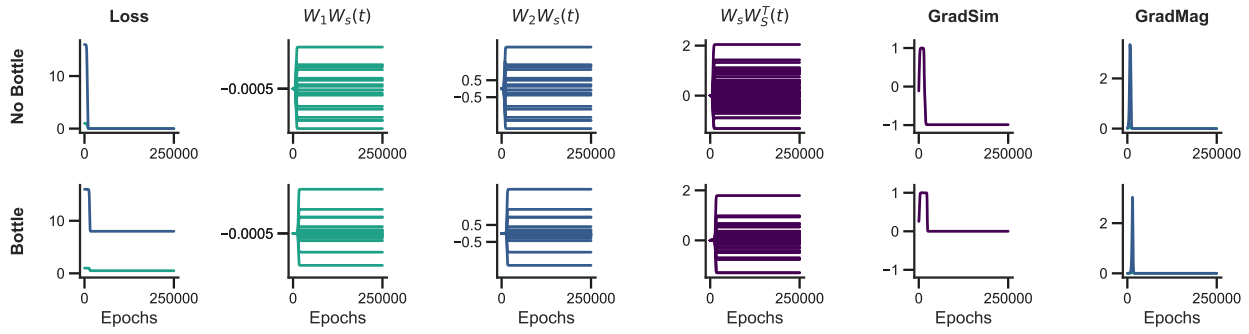

Figure 9: Zero balanced; $N_1 = 10, N_S = 1, N_{o2} = 2 = N_{o1}$. Aligned tasks, different scale $\Delta M_{12} = 0.25$.

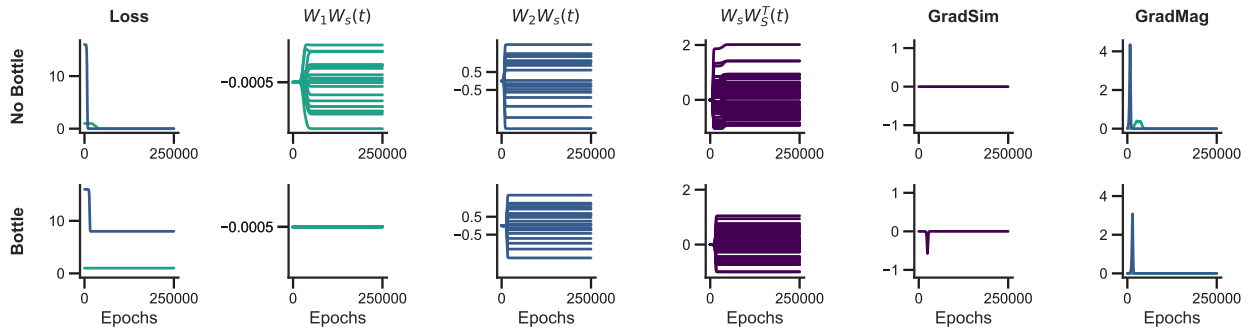

Figure 10: Non-Bottlenecked $N_1 = 10, N_S = 4, N_{o2} = 2 = N_{o1}$ (top) versus Bottlenecked $N_1 = 10, N_S = 1, N_{o1} = 2, N_{o2} = 2$ (bottom) networks trained on orthogonal tasks with a different scale $\Delta M_{12} = 0.25$.

Fully Aligned Tasks. First, we consider zero-balanced networks trained on a bottleneck network. Apart from faster convergence, the bottleneck case does not seem to be different, Figure 9. Idem for λ-balancedness. This is not surprising, since the high alignment reduces the actual rank, so the network is not really bottlenecked.

Fully Orthogonal Tasks. Due to the bottleneck, one task fails to be learned. This raises the question of whether it is possible to predict which task will be suppressed based on the spectrum $\mathbf{S}_\xi$. Figure 10 illustrates that the bottleneck enforces simultaneous learning even for orthogonal tasks, effectively coupling their updates and driving the suppressed task away from its optimal direction, leading to suboptimal convergence. This effect is further reflected in the emergence of negative cosine similarity between task gradients. When tasks differ in scale, we observe that the task with larger magnitude dominates, while the lower-scale task fails to learn.

C.8 Detailed discussion on remaining assumptions.

While we acknowledge the limitations of our assumptions, we note that these assumptions are standard in theory and provide practically relevant insights.

A. Linear Networks. As highlighted in the position paper by Nam et al. (2025) , linear networks are often dismissed as lacking expressive power. However, despite their apparent simplicity, their training dynamics are inherently non-linear. This non-linearity allows them to capture and explain a range of key phenomena observed in modern deep learning, e.g., neural collapse (Papyan et al., 2020), and grokking (Kunin et al., 2024).

B. Whitened Inputs (Assumption A1). Although strong, the assumption of whitened inputs is a standard analytical simplification in theoretical deep learning, enabling exact solutions for network dynamics

(e.g., Saxe et al. (2014); Braun et al. (2022); Kunin et al. (2024)). Relaxing this assumption typically prevents closed-form analysis.

Importantly, empirical work supports the benefits of decorrelated inputs: e.g, decorrelated backpropagation ((Dalm et al., 2025)) improves learning efficiency, while sparse autoencoders demonstrate that decorrelated representations improve downstream performance (Saraswatula & Klindt, 2025). Moreover, we verify robustness by removing the whitening assumption in experiments (e.g., Figure 15).

C.9 Task Alignment and the Shared Spectrum

Consider the combined task matrix $\xi = \begin{pmatrix} \mathbf{\Sigma}_{1o} \\ \mathbf{\Sigma}_{2o} \end{pmatrix}$ with ordered SVD $\xi = \mathbf{U}_\xi \mathbf{S}_\xi \mathbf{V}_\xi^\top$, and the task SVDs $\mathbf{\Sigma}_{1o} = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^\top$ and $\mathbf{\Sigma}_{2o} = \mathbf{U}_2 \mathbf{S}_2 \mathbf{V}_2^\top$. Since $\xi^\top \xi = \mathbf{\Sigma}_{1o}^\top \mathbf{\Sigma}_{1o} + \mathbf{\Sigma}_{2o}^\top \mathbf{\Sigma}_{2o}$, expanding both sides in their SVDs gives

$$\mathbf{V}_\xi \, \mathbf{S}_\xi^2 \, \mathbf{V}_\xi^\top = \mathbf{V}_1 \mathbf{S}_1^2 \mathbf{V}_1^\top + \mathbf{V}_2 \mathbf{S}_2^2 \mathbf{V}_2^\top \; = \xi^\top \xi \equiv \; \mathbf{M}. \tag{150}$$

The combined singular values are therefore $s_\xi^i = \sqrt{\lambda_i}$, where λ_i are the eigenvalues of $\mathbf{M}$, so the spectrum $\mathbf{S}_\xi$ reflects how the two tasks' singular modes combine.

For perfectly aligned tasks, $\mathbf{V}_1 = \mathbf{V}_2 = \mathbf{V}$, so $\mathbf{M} = \mathbf{V}(\mathbf{S}_1^2 + \mathbf{S}_2^2)\mathbf{V}^\top$ and $\lambda_i = s_{i,1}^2 + s_{i,2}^2$: each combined mode pairs the two tasks' i-th singular values in quadrature. For completely orthogonal tasks, $\text{span}(\mathbf{V}_1) \perp \text{span}(\mathbf{V}_2)$, so $\mathbf{V}_1^\top \mathbf{V}_2 = \mathbf{0}$ and

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{pmatrix} \begin{pmatrix} \mathbf{S}_1^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2^2 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{pmatrix}^\top, \tag{151}$$

so $\{\lambda_i\} = \{s_{1,1}^2, \ldots, s_{r_1,1}^2, \; s_{1,2}^2, \ldots, s_{r_2,2}^2\}$, the ordered union of the two task spectra.

When the tasks are conflicting, the way the combined singular values decompose into the task-specific ones is more involved. For two one-dimensional tasks with $s_1 = s_2 = s$ and overlap α (definition 2.1), the combined spectrum is

$$\mathbf{S}_\xi = \{ \, s\sqrt{1+\alpha}, \; s\sqrt{1-\alpha} \, \}, \tag{152}$$

a stronger *cooperative* mode and a weaker *conflicting* mode. Increasing α widens the gap between them, strengthening the cooperative mode and weakening the conflicting one. We discuss the impact of this gap on the learning dynamics in Appendix D.1 (Figure 13).

Impact of task imbalance on the shared spectrum If we additionally consider the case where $s_1 \neq s_2$ we can distinguish the effect of misalignment from effects coming from different mode strengths between the tasks. For two one-dimensional tasks with overlap α, the combined eigenvalues are

$$\lambda_\pm = \tfrac{1}{2}\Big[(s_1^2 + s_2^2) \pm \sqrt{(s_1^2 - s_2^2)^2 + 4 s_1^2 s_2^2 \alpha^2}\Big], \tag{153}$$

so the gap between the two combined modes is $\sqrt{(s_1^2 - s_2^2)^2 + 4 s_1^2 s_2^2 \alpha^2}$, with one term set by the magnitude imbalance $(s_1^2 - s_2^2)$ and the other by the alignment α. When the magnitudes are balanced this reduces to the $s\sqrt{1 \pm \alpha}$ split above; when they are imbalanced, the $(s_1^2 - s_2^2)^2$ term dominates and the gap is governed mainly by the magnitude difference.

The imbalance also changes how each task loads onto the combined modes. Projecting the task directions onto $\mathbf{V}_\xi$ shows that the stronger task is carried almost entirely by the strong (fast) mode, while the weaker task retains a substantial component on the weak (slow) mode. At $\alpha = 0.92$ with $s_1{:}s_2 \approx 1{:}4$, the loadings on the (fast, slow) modes are [1.00, 0.02] for the strong task and [0.93, 0.37] for the weak task: although both tasks are nearly aligned and begin learning together along the fast cooperative mode, only the weaker task carries appreciable weight on the slow mode and must wait for it to resolve. At low alignment ($\alpha = 0.24$) the split is sharper, with loadings [1.00, 0.02] for the strong task and [0.26, 0.97] for the weak task: here each task occupies a different mode, as the near-orthogonal geometry separates them in addition to the magnitude imbalance. In both cases the stronger task converges first, and the resulting asymmetry in learning speed is driven by the magnitude imbalance rather than by misalignment (Appendix D.1).

D Experiments

Most experiments were conducted on Google Colab and a local machine. Synthetic-data simulations were run exclusively on CPU, without GPU acceleration, using a local system with an Intel Core i7-1165G7 processor (11th Gen, 4 cores, 8 threads, 2.80 GHz base frequency); these toy-data simulations complete in under 10 minutes. Experiments on MNIST and NYUv2 were run on an HPC cluster with A100 (Ampere) GPUs (up to 40 GB memory) and 2x 16-core AMD EPYC 7282 CPUs. All code was implemented in Python using the PyTorch framework. For Multi-MNIST, we used a subset of 50,000 images to reduce training time. The corresponding code repository is included in the Supplemental Material and will be made publicly available on GitHub upon acceptance, including a notebook to reproduce the figures in the main paper. We additionally provide text files with the tracked statistics from the Multi-MNIST and NYUv2 experiments; these can be loaded into the corresponding notebook to reproduce the corresponding figures.

D.1 γ-MTL tasks

In this section, we explain how the tasks used for the results on Figure 3 were created. Additionally, we present plots for different configurations of these type of tasks.

Generate an orthonormal basis $\mathbf{R}$ and choose B_1 as the first N_1 rows, $B_1^{\perp}$ as the remaining rows. Then we can construct two tasks like,

$$\mathbf{Y}_1 = \alpha_1 \mathbf{B}_1 \mathbf{X} \tag{154}$$

and construct

$$\mathbf{Y}_2 = \alpha_2 [(\gamma * \mathbf{B}_1 + (1-\gamma)\mathbf{B}_1^{\perp}]\mathbf{X}. \tag{155}$$

As such γ controls the alignment of the two tasks ($\gamma = 1$, perfect alignment, $\gamma = 0$, perfect orthogonality). While α_i only alters the relative magnitude between the tasks.

D.1.1 Additional experiments with γ-tasks

We compare the analytical predictions based on Theorem 3.2 with the numerical simulations in Figure 3 in the main paper to analyse the impact of task alignment on the learning dynamics.

Balanced magnitudes. While Figure 3 presents examples of imbalanced tasks trained on a linear network, Figure 11 shows the dynamics for tasks with balanced magnitudes. We discuss the exact impact in more detail below.

Non-Linear Networks. Figure 12 illustrates the corresponding dynamics when the same tasks are trained using a ReLU network. While the exact shape of the network functions and losses are different when the tasks are trained on a nonlinear network, our analyses presented in Section 4 remains valid.

Intermediate α's. To understand how the dynamics change with the degree of conflict $0 < \alpha < 1$, we plot the analytical predictions and numerical simulations of the task-specific losses, network functions, gradient similarities, and gradient magnitudes (Figure 13). We consider two tasks with $N_1 = 10$ and $N_{o_1} = N_{o_2} = 1$ at $\alpha \in \{0.92, 0.71, 0.24\}$, under both equal and imbalanced task magnitudes, in order to separate the effects of misalignment and magnitude imbalance.

The first row shows $\alpha = 0.92$, with combined spectrum $\mathbf{S}_\xi = [1.39, 0.28]$. The strong cooperative mode dominates, so both tasks learn nearly simultaneously, closely resembling fully aligned tasks. The second, conflicting mode is much weaker ($s_\xi = 0.28$): it resolves slowly and is still evolving late in training, visible as the late drift in the network functions (Figure 13**B-C**). As the alignment decreases to $\alpha = 0.71$ (middle row), the gap between the two mode strengths narrows, $\mathbf{S}_\xi = [1.31, 0.54]$, and both modes become visible: the losses decrease in two stages, the first accompanied by positive gradient similarity (the cooperative mode) and the second by negative similarity (the conflicting mode). Finally, at $\alpha = 0.24$ (bottom row) the spectrum is $\mathbf{S}_\xi = [1.11, 0.87]$ and the gap is smallest, so the two modes have nearly equal strength: the cooperative and conflicting phases follow closely one after the other, with only a brief positive similarity before the dynamics

turn negative. Interestingly, the more aligned tasks (higher α) produce the deepest and longest-lasting dip (for non-zero gradient magnitude) in the gradient cosine similarity. This may seem counterintuitive, but follows from the separation in time between the cooperative and conflicting modes. Their resolution times differ by a factor $\sqrt{(1+\alpha)/(1-\alpha)}$: at high α the conflicting mode is weak ($s_\xi = s\sqrt{1-\alpha}$) and therefore slow, resolving long after the cooperative mode has saturated. The late-stage gradient is then almost purely anti-aligned, driving the cosine similarity close to -1. For less aligned tasks the conflicting mode is stronger and faster, resolving concurrently with the cooperative mode, so the gradient remains a mixture of cooperative and conflicting components and the negative similarity is partially cancelled.

Effect of magnitude imbalance. Figure 14 repeats the same alignment sweep ($\alpha \in \{0.92, 0.71, 0.24\}$) with imbalanced task magnitudes ($\Delta M = 0.2$). The combined spectra are now $\mathbf{S}_\xi = [4.11, 0.38]$, $[4.06, 0.70]$, and $[4.01, 0.97]$: the strong mode is pinned near 4 across all three rows and barely varies with α, since the gap $\sqrt{(s_1^2 - s_2^2)^2 + 4s_1^2 s_2^2 \alpha^2}$ (Section D.1) is dominated by the magnitude difference rather than the alignment. The alignment effect on the spectrum is therefore largely masked, and the three rows resemble one another more closely than in the balanced case (Figure 13). The imbalance instead appears as a clear asymmetry between the tasks: the stronger task (blue) converges almost immediately, its loss reaching zero early (**A**), while the weaker task (green) lags. This asymmetry is driven by the magnitude difference rather than by misalignment. At $\alpha = 0.92$, where the tasks are nearly aligned, both project mainly onto the fast (strong) mode (we can measure thuis by projecting each task's singular vector onto $\mathbf{V}_\xi$, yielding $[1.00,\ 0.02]$ for the strong task and $[0.93,\ 0.37]$ for the weak task), so both begin learning together along the cooperative direction. The stronger task is carried almost entirely by this fast mode and converges early, while the weaker task retains a residual component on the slow mode (0.37) and must wait for it to resolve, producing the loss asymmetry in (**A**) even though the tasks are well aligned. The imbalance is also visible in the gradient diagnostics, as a large early spike and strongly unequal task contributions in the gradient magnitudes (**E**) absent under balanced magnitudes. The weaker mode still tracks the alignment ($s_\xi = 0.38, 0.70, 0.97$ as α decreases), so misalignment continues to set the conflicting-mode strength; it is the relative effect of α that the imbalance suppresses.

Non-Whitened Input Data. Figure 15 illustrates the difference between the analytical predictions for whitened input with numerical simulations on non-whitened input data for zero and λ-balanced initialisations.

D.2 Higher-dimensional output γ-tasks

To probe intermediate and mixed regimes, we construct pairs of linear teacher tasks whose overlap matrix $\mathbf{O}_{12}$ (Definition 2.1) has a singular-value spectrum $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$, with each $\alpha_k \in [0, 1]$. This extends the previous data generation process to allow us to build mixed spectra containing all three alignment categories simultaneously. We discuss the resulting analytical and numerical simulations below.

Data Generation. Let N_1 be the input dimension and draw a random orthonormal basis $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_{N_1}] \in \mathbb{R}^{N_1 \times N_1}$ (from the QR decomposition of a Gaussian matrix). We split its first $2d$ columns into two mutually orthonormal blocks,

$$\mathbf{A} = [\mathbf{q}_1, \dots, \mathbf{q}_d], \qquad \mathbf{B} = [\mathbf{q}_{d+1}, \dots, \mathbf{q}_{2d}], \tag{156}$$

which requires $2d \leq N_1$, i.e. $N_{o_1} + N_{o_2} \leq N_1$, the admissibility condition under which the overlap spectrum is well defined (Section 2.2). The two teachers are then

$$\mathbf{W}_1 = \mathbf{A}^T, \qquad \mathbf{W}_2 = \operatorname{diag}(\boldsymbol{\alpha})\, \mathbf{A}^T + \operatorname{diag}\big(\sqrt{1-\boldsymbol{\alpha}^2}\big)\, \mathbf{B}^T, \tag{157}$$

so that the k-th mode of task 2 is $\mathbf{w}_2^k = \alpha_k\, \mathbf{a}_k + \sqrt{1-\alpha_k^2}\, \mathbf{b}_k$, a unit vector at angle $\arccos \alpha_k$ from the corresponding mode $\mathbf{a}_k$ of task 1. Inputs $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$ are whitened so that $\boldsymbol{\Sigma}_{11} = \mathbf{I}$, and the targets are $\mathbf{Y}_i = c_i\, \mathbf{W}_i \mathbf{X}^T$, where c_i set the relative task magnitudes.

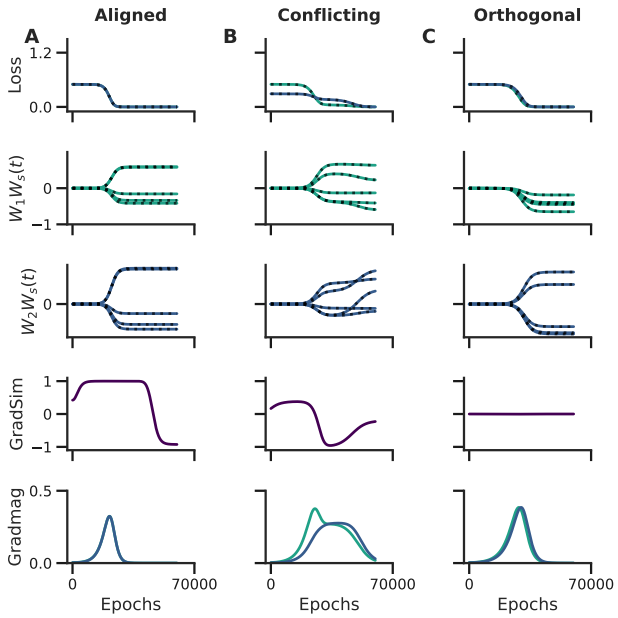

Figure 11: **Top three rows**: Analytical predictions (dots) versus numerical simulations of task-specific losses (task 1: green; task 2: blue) and network functions for a linear neural network ($N_1 = 5, N_S = 2, N_{oi} = 1$) trained on fully aligned (A), conflicting (B), and orthogonal tasks (C) with magnitudes balanced $\Delta M_{1,2} = 1$. **Bottom two rows**: The corresponding cosine similarity between task-specific gradients (GradSim) and gradient magnitudes (GradMag) of both tasks.

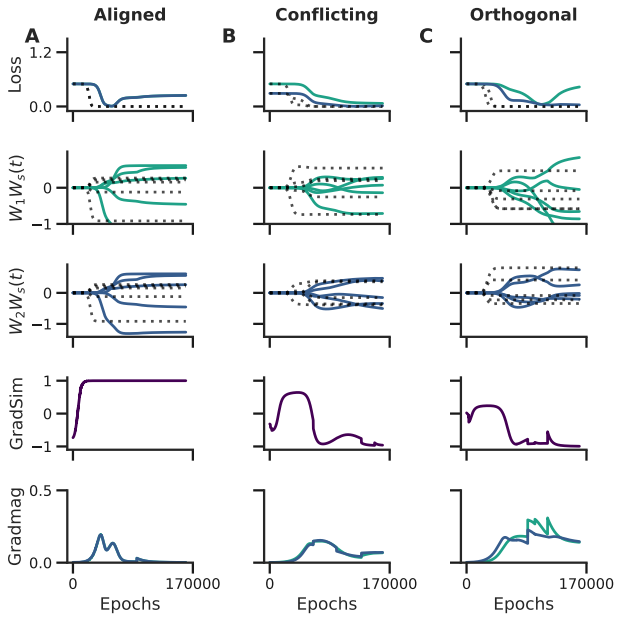

Figure 12: **Top three rows**: Analytical predictions for two-linear networks (dots) versus numerical simulations of task-specific losses (task 1: green; task 2: blue) and network functions for a ReLU neural network ($N_1 = 5, N_S = 2, N_{oi} = 1$) trained on fully aligned (A), conflicting (B), and orthogonal tasks (C) with magnitudes balanced $\Delta M_{1,2} = 1$. **Bottom two rows**: The corresponding cosine similarity between task-specific gradients (GradSim) and gradient magnitudes (GradMag) of both tasks.

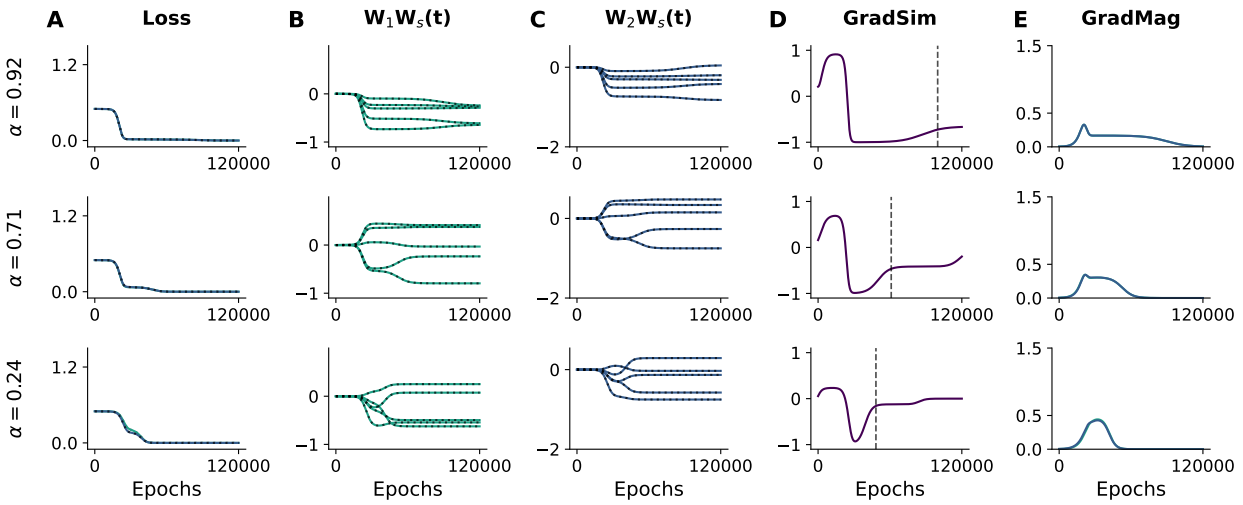

Figure 13: **Top three rows**: Analytical predictions (dots) versus numerical simulations of task-specific losses (task 1: green; task 2: blue) and network functions for a linear neural network with ($N_1 = 5, N_S = 2, N_{oi} = 1$) trained on conflicting tasks with alignment corresponding to singular values of the overlap matrix equal to ($\alpha = 0.97, \alpha = 0.71, \alpha = 0.24$) balanced magnitudes $\Delta M_{1,2} = 1$. **Bottom two rows**: The corresponding cosine similarity between task-specific gradients (GradSim) and gradient magnitudes (GradMag) of both tasks. We plot a vertical line in column D corresponding to the step where the losses reach 10^{-4} to differentiate between numerically noisy signals of the gradient similarities.

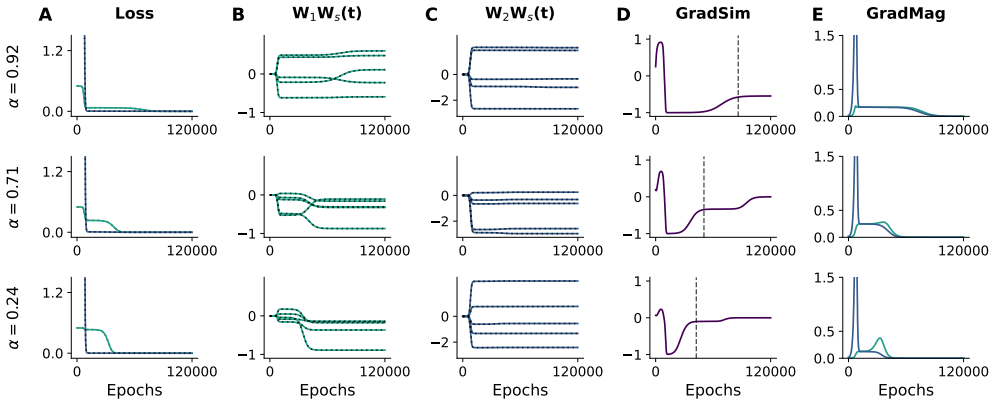

Figure 14: **Top three rows**: Analytical predictions (dots) versus numerical simulations of task-specific losses (task 1: green; task 2: blue) and network functions for a linear neural network with ($N_1 = 5, N_S = 2, N_{oi} = 1$) trained on conflicting tasks with alignment corresponding to singular values of the overlap matrix equal to ($\alpha = 0.97, \alpha = 0.71, \alpha = 0.24$) imbalanced magnitudes $\Delta M_{1,2} = 0.2$. **Bottom two rows**: The corresponding cosine similarity between task-specific gradients (GradSim) and gradient magnitudes (GradMag) of both tasks. We plot a vertical line in column D corresponding to the step where the losses reach 10^{-4} to differentiate between numerically noisy signals of the gradient similarities.

D.2.1 Simulations on mixed Aligned Spectra

We analyse different mixed alignment regimes to understand how to move beyond one-dimensional tasks. Figure 16 fixes a shared aligned core ($\alpha_1 = \alpha_2 = 1$) and varies the remaining two modes from mildly conflicting ($[0.9, 0.75]$, top row) to strongly conflicting ($[0.3, 0.2]$, middle row) to orthogonal ($[0, 0]$, bottom row), at equal task magnitudes ($s_1 = s_2$). We compare the corresponding plots to Figure 17, where we plot the same pairs of tasks with imbalanced task magnitudes.

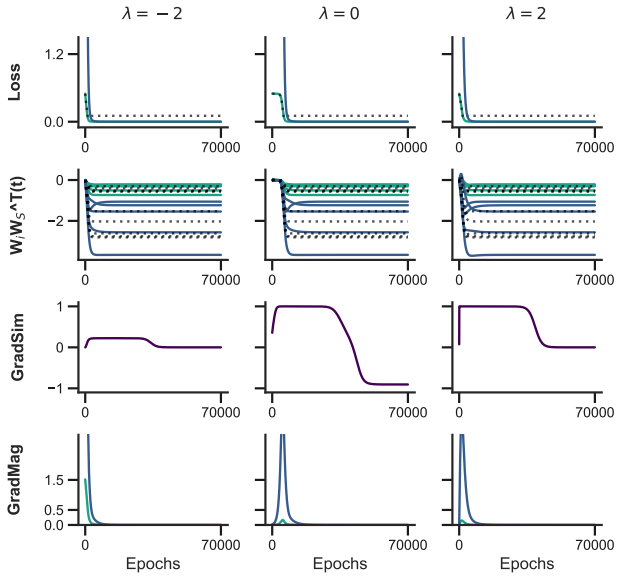

Figure 15: Comparison of analytical predictions versus numerical simulations of task-specific losses and network functions for a Two-layer Linear network trained on Non-Whitened Aligned tasks ($N_1 = 5, N_S = 2, N_{oi} = 1$) for zero-balanced and lambda-balanced initialisations.

Losses and network functions. In all three rows the loss (**A**) undergoes a fast initial drop driven by the two aligned modes, which are the strongest combined modes, and dominate the early descent. The secondary block has little effect on this initial drop but does shape the slow tail: the mildly conflicting row converges slowest, since its conflicting modes are weak and therefore resolve slowly, leaving a long residual tail, whereas the stronger secondary modes of the other rows resolve faster. The two task losses remain close within each row, consistent with the balanced magnitudes, though they need not coincide exactly: in the orthogonal row a small offset remains, as the two tasks' modes are private and resolve at slightly different times according to their individual strengths. The network functions (**B**-**C**) make the secondary modes visible.

Gradient similarity and magnitude. The secondary block (the two smallest α_i) is most visible in the gradient diagnostics (**D**-**E**). All three rows begin with a strong positive gradient similarity from the shared aligned modes and differ only in what follows, which we interpret in terms of the strength and timing of the conflicting modes. For the mildly conflicting row ($[0.9, 0.75]$), the gradient similarity reaches the deepest and longest-lasting negative valu, which may seem counterintuitive. We attribute this to the weakness of its conflicting modes: since $s_\xi = s\sqrt{1-\alpha}$ is small, these modes are slow and resolve only after the cooperative modes have saturated, so the late-stage gradient consists almost entirely of the anti-aligned residual and the cosine approaches -1. For the strongly conflicting row ($[0.3, 0.2]$), the conflicting modes are stronger and faster, resolving concurrently with the cooperative ones; the gradient then remains a mixture of cooperative and conflicting components, and the negative excursion is correspondingly shallower. In both rows the similarity becomes negative while the gradient magnitude is still nonzero (**E**), which we take as evidence that the negative cosine reflects genuine conflict rather than a numerical effect. The orthogonal row ($[0, 0]$) behaves differently: after the cooperative phase the gradient similarity remains close to zero (until the losses became zero), consistent with the two tasks' gradients being mutually orthogonal.

D.2.2 Simulations on mixed Orthogonal Spectra

To contrast the role of early aligned modes with that of weakly aligned or orthogonal ones, we plot the learning dynamics for conflicting-orthogonal ($\alpha = [0.3, 0.2, 0, 0]$) and fully orthogonal ($\alpha = [0, 0, 0, 0]$) tasks in Figure 18, at balanced magnitudes.

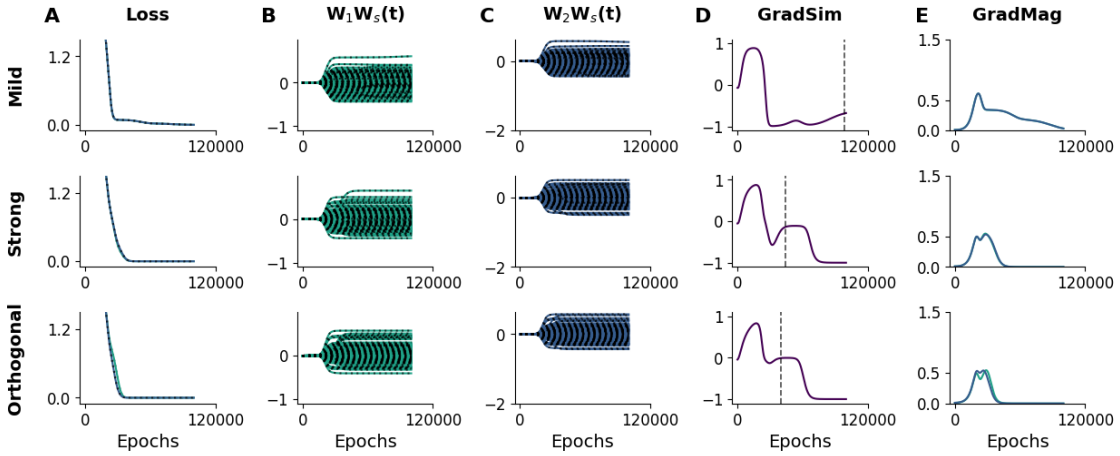

Figure 16: Analytical solutions (black dotted lines) match numerical simulations of the task-specific losses (task 1: green; task 2: blue solid lines) and, **columns B, C**, the components of the network functions $\mathbf{W}_i\mathbf{W}_S$, for a linear neural network ($N_1 = 20$, $N_S = 8$, $N_{o_1} = N_{o_2} = 4$). The network is trained on aligned ($\alpha = [1, 1, 0.9, 0.75]$, top row), conflicting ($\alpha = [1, 1, 0.3, 0.2]$, middle row), and orthogonal ($\alpha = [1, 1, 0, 0]$, bottom row) regression tasks constructed as described in Appendix D.1, with balanced magnitudes. **Columns D, E.** Corresponding cosine similarity between task-specific gradients (GradSim, Eq. 16) and gradient magnitudes (GradMag, Eq. 15) for both tasks over training until convergence. The vertical dotted line (column D) indicates the training step at which the losses became zero.

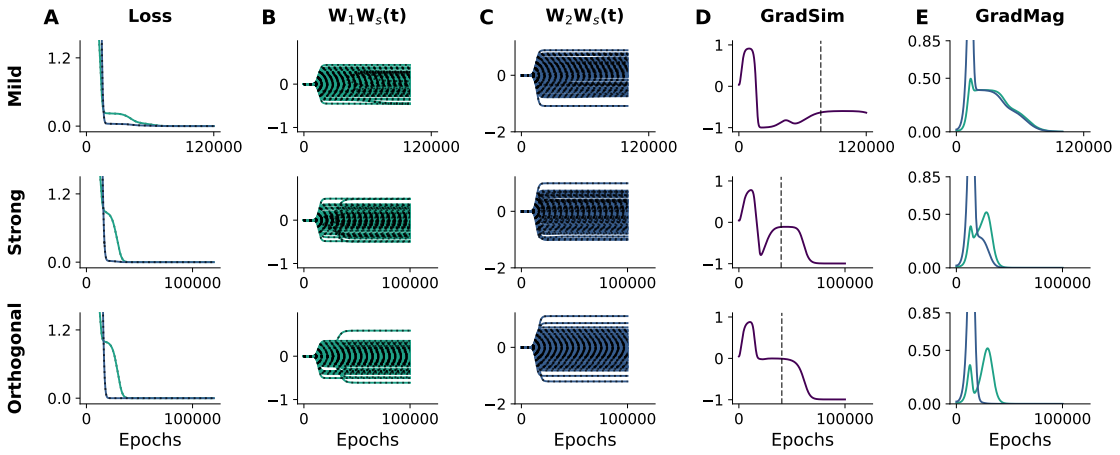

Figure 17: As in Figure 16, but with magnitude imbalance $\Delta M = 0.2$. Unlike the balanced case, the imbalance introduces a clear task asymmetry: in every row the stronger task (blue) converges first while the weaker task (green) lags, visible as staggered task losses (**A**) and unequal-height gradient magnitude bumps (**E**) rather than the coincident curves of the balanced setting.

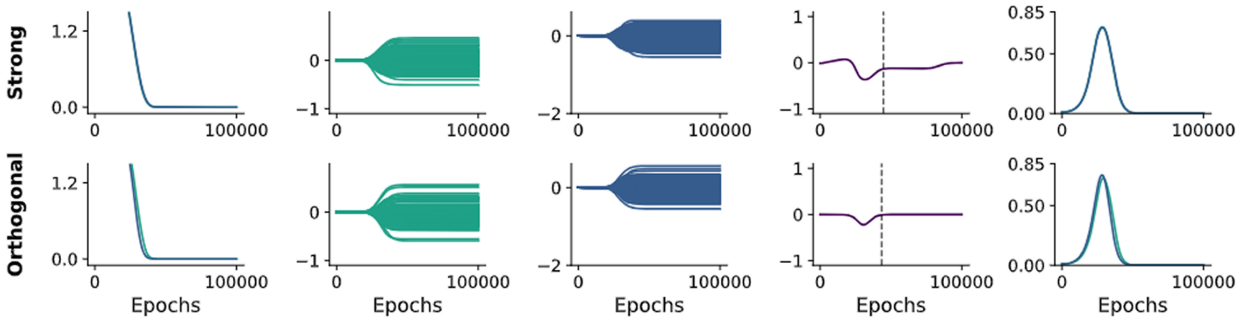

Figure 18: Numerical simulation of the learning dynamics of a two-layer linear network ($N_1 = 20$, $N_S = 8$, $N_{o_1} = N_{o_2} = 4$) trained on (top) conflicting-orthogonal ($\alpha = [0.3, 0.2, 0, 0]$) and (bottom) fully orthogonal ($\alpha = [0, 0, 0, 0]$) tasks, with balanced magnitudes.

Losses and network functions. In the fully orthogonal row the two task losses (Figure 18**A**) resolve in sequence rather than together, with a visible offset between them. We attribute this to the structure of the combined spectrum: for orthogonal tasks every mode is private to a single task (the combined spectrum is the union of the two task spectra), so modes of slightly different strength resolve at slightly different times and the two losses separate, with no mechanism coupling them. In the conflicting-orthogonal row the losses overlap more closely, which we attribute to its conflicting modes: these are shared between the tasks and resolve on a common timeline, synchronizing the two losses even as they oppose the tasks' directions.

Cosine similarity and gradient magnitudes. In the conflicting-orthogonal row ($\alpha = [0.3, 0.2, 0, 0]$), the absence of an aligned mode removes the strong early cooperation: the gradient similarity barely peaks positive, dips mildly negative as the conflicting modes resolve, and then settles toward zero, as the two orthogonal modes contribute no similarity and dilute the conflict. In the fully orthogonal row ($\alpha = [0, 0, 0, 0]$), the gradient similarity stays near zero throughout active learning: the two tasks' gradients are mutually orthogonal, so there is neither cooperation nor conflict. In both rows the gradient magnitudes (Figure 18**E**) form a single bump (due to the very similar mode strengths), but the two tasks coincide more closely in the conflicting-orthogonal row than in the fully orthogonal one: the shared conflicting modes synchronize the tasks, whereas in the fully orthogonal case each task's private modes resolve at slightly different times, slightly offsetting their magnitude curves.

D.2.3 Representations at convergence

In Figure 19, we extend the analysis of the learned representations from Section 5.1 to higher dimensions and mixed spectra. We construct two task pairs: an aligned, mildly conflicting one ($\alpha = [1, 0.8, 0.75, 0.68]$) and an aligned, strongly conflicting one ($\alpha = [1, 0.2, 0.14, 0.12]$). We compare the representations when task magnitudes are balanced (Figure 19**A**) and imbalanced (Figure 19**B**)

Balanced Tasks. For the mildly conflicting pair (Figure 19**A**(c, d)), the combined representation $\mathbf{U}_\xi \mathbf{S}_\xi \mathbf{U}_\xi^\top$ differs markedly from the single-task one: its diagonal entries are weaker ($\mathbf{\Delta USU}^\top < 0$ on the diagonal, as the shared modes redistribute strength between the tasks), and pronounced off-diagonal entries appear, reflecting correlations between the two tasks' output dimensions introduced by their shared directions. For the strongly conflicting pair (Figure 19(e, f)), the representation is instead almost identical to the single-task case ($\mathbf{\Delta USU}^\top \approx 0$), both on and off the diagonal, except along the directions that remain strongly aligned, in particular the shared mode $\alpha_1 = 1$. The weakly aligned directions ($\alpha_i \leq 0.2$) leave the representation essentially unchanged. On Figure 19**C** We plot the corresponding $\Delta\mathbf{VSV}^\top$, which sows the same dependence; it deviates strongly for the mildly conflicting pair (**a**), whose shared input directions introduce structure absent from the block-diagonal direct sum, and is close to zero for the strongly conflicting pair (**b**), apart from the residual contribution of the aligned mode, indicating that the tasks remain nearly separable in input space.

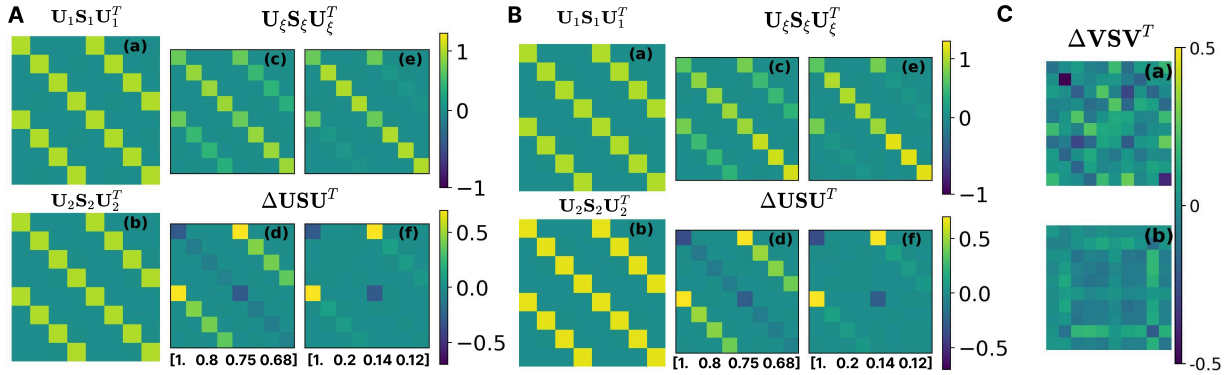

Figure 19: **A**.Output representations for two mixed-spectrum task pairs at balanced magnitudes ($s_1 = s_2$): a high-alignment spectrum $\alpha = [1, 0.8, 0.75, 0.68]$ (**c**, **d**) and a low-alignment spectrum $\alpha = [1, 0.2, 0.14, 0.12]$ (**e**, **f**). Panels **a**, **b** show the single-task RSMs $\mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top$, **c**, **e** the combined RSM $\mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^\top$, and **d**, **f** their difference $\mathbf{\Delta USU}^\top$. The diagonal of $\mathbf{\Delta USU}^\top$ gives each mode's change in strength under joint training, and the off-diagonal entries the correlations introduced between the two tasks' output classes. Both are pronounced under high alignment (**d**) and vanish under low alignment (**f**), where only the aligned modes deviate and the weakly aligned directions leave the representation unchanged ($\mathbf{\Delta} \approx 0$). **B** Same but with $\Delta M = 2$.

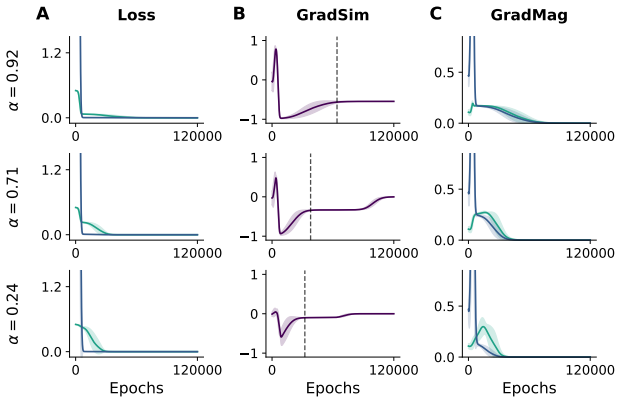

Figure 20: Repeating Figure 14 for 5 seeds to validate consistent behaviour among different seeds

Task Imbalance. Figure 19**B** is essentially constructed in the same way as Figure 19**A** but with imbalanced task magnitudes ($s_1 \neq s_2$). The single-task RSMs $\mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top$ (**a**, **b**) now differ in strength, reflecting the imbalance. As in the balanced case, the difference $\mathbf{\Delta USU}^\top$ (**d**, **f**) deviates strongly under high alignment ($\alpha = [1, 0.8, 0.75, 0.68]$) and only at the aligned modes under low alignment ($\alpha = [1, 0.2, 0.14, 0.12]$, $\mathbf{\Delta} \approx 0$ elsewhere), confirming that the alignment-driven structure is preserved under magnitude imbalance, with the imbalance adding an asymmetry between the two tasks' modes.

D.2.4 Consistency across seeds

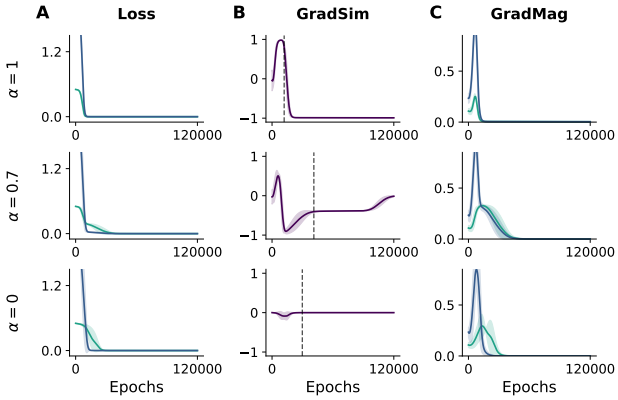

Figure 21: Repeating Figure 3 for 5 seeds to validate consistent behaviour among different seeds

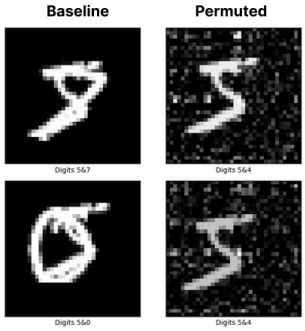

Figure 22: Two samples of the Baseline and Permuted Multi-MNIST datasets before whitening X.

D.3 Multi-MNIST

Multi-MNIST is a multi-task extension of the original MNIST dataset Lecun et al. (1998), introduced by Sabour et al. (2017) and widely used as a benchmark in the multi-task learning (MTL) literature. It is constructed by overlaying pairs of digits drawn from the same split (training or test) but belonging to different classes. Each digit is independently shifted by up to four pixels in both horizontal and vertical directions, resulting in images of size 36×36=36×36, which are flattened into 1296-dimensional input vectors and subsequently whitened.

The dataset is publicly available via Papers with Code (`https://paperswithcode.com/paper/dynamic-routing-between-capsules`) and released under a CC-BY-SA license. In our MTL setting, each image defines two tasks: classification of the left digit and classification of the right digit.

We consider several adaptations of Multi-MNIST that vary the degree of digit overlap and the alignment of task representations. Throughout this section, we refer to left-digit classification as Task 1 and right-digit classification as Task 2.

In this section, we discuss the classic Multi-MNIST baseline and a permuted multi-MNIST version. Two samples of each dataset are visualised on Figure 22. Finally, we include experiments on deeper networks (Figure 25) and non-uniform task loss weights (Figure 15).

D.3.1 Baseline Multi-MNIST

In this setup we do not alter the overlap between the left and right digit and keep the configuration the dataset as introduced by Sabour et al. (2017). We constructed the baseline Multi-MNIST tasks by sampling pairs of MNIST digits and placing them on a shared 36 × 36 canvas. Each example consists of two independently

chosen digits from different classes, each originally 28×28, inserted at random spatial offsets within the larger canvas. The two task labels are the class identities of the first and second digit, respectively, encoded as one-hot vectors. The observed input is formed by taking the pixelwise maximum of the two single-digit canvases, yielding a superposed two-digit image. The resulting 36×36 image is then flattened into a 1296-dimensional vector and used as the common input for the two-task model.

Task Alignment. To compute the alignment for the Baseline Multi-Mnist tasks, the singular values of the overlap matrix are averaged over several seeds and summarised in Table 1.

Table 1: Alignment singular values α_i for the Baseline Multi-MNIST configuration, averaged over 3 seeds.

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
α_i	1.000	0.925	0.899	0.893	0.850	0.827	0.784	0.755	0.691	0.526
$\pm$ std	0.000	0.005	0.002	0.003	0.004	0.001	0.015	0.004	0.020	0.025

By Definition 2.1, the tasks are technically conflicting, since $0 < \alpha_i < 1$ for all but the first direction. However, $\alpha_i \geq 0.69$ for the first nine directions, with only the smallest mode dropping to $\alpha_{10} \approx 0.53$ (Table 1), so the tasks are in practice predominantly aligned. From the analysis in the main paper and Figure 16(top tow), we therefore expect the tasks to converge nearly simultaneously. This is confirmed by the experiments in Figure 5, where we compare the task-specific losses obtained from training a linear MTL network and a nonlinear (ReLU) MTL network. The gradient cosine similarity reflects this graded alignment: it rises sharply early in training, when the dominant, strongly aligned modes dominate the loss landscape and both tasks' gradients point in nearly the same direction direction; as this mode saturates, learning becomes driven by the remaining, only partially aligned modes, producing a transient negative dip in cosine similarity as the gradients diverge to resolve these weaker, task-specific directions.

Concerning the RSMs, Figure 5 shows a similar analysis to Figure 4 to compare the RSM's of the input and output data for the single tasks (i.e., only the left or right digit estimation) and find that our analyses generalise well to the real-world dataset. The combined RSMs of the input data yield weaker structural relationships, and the combined data RSMs of the input data create new structural relationships. In Figure 23 compare these predictions to the actual neural representations after training a neural network with $N_S = 20$ during 25000 epochs ($\eta = 0.01$). We see that the neural network captures weaker structural relationships for the output while finding new relationships (although weak) for the input data.

D.3.2 Permuted Multi-MNIST

To study the impact of task alignment, we modify the original Multi-MNIST dataset to construct a pair of tasks with a different degree of alignment relative to the Baseline configuration. For the Permuted version, we generate the same paired two-digit inputs as in the Baseline setting, but additionally apply a fixed random orthogonal transformation to the second digit canvas before combining it with the first. Concretely, after placing the second digit on its 36×36 canvas, we transform that canvas by $\mathbf{R}\ \mathrm{X}\ \mathbf{R}^\top$, where $\mathbf{R}$ is a random orthogonal matrix sampled once for the dataset generator. The two canvases are then merged by a pixelwise maximum as before. This preserves the overall two-task structure while disrupting the spatial correspondence between the second task and the original input coordinates, thereby creating a misaligned task pair.

Task alignment The singular values of the overlap matrix are averaged over several seeds and summarised in Table 2. By Definition 2.1, the Permuted tasks are also technically conflicting, since $0 < \alpha_i < 1$ for all but the first direction. Unlike the Baseline configuration, however, α_i drops sharply after the first mode, with $\alpha_2, \ldots, \alpha_{10} \leq 0.24$ (Table 2), so the tasks are mostly conflicting and closer to orthogonal beyond a single shared direction. From the analysis in the main paper and Figure 16(bottom row), we therefore expect a brief initial period of shared learning along the single aligned mode, followed by sustained gradient conflict along the remaining directions. This is consistent with the experiments in Figure 5: both tasks begin

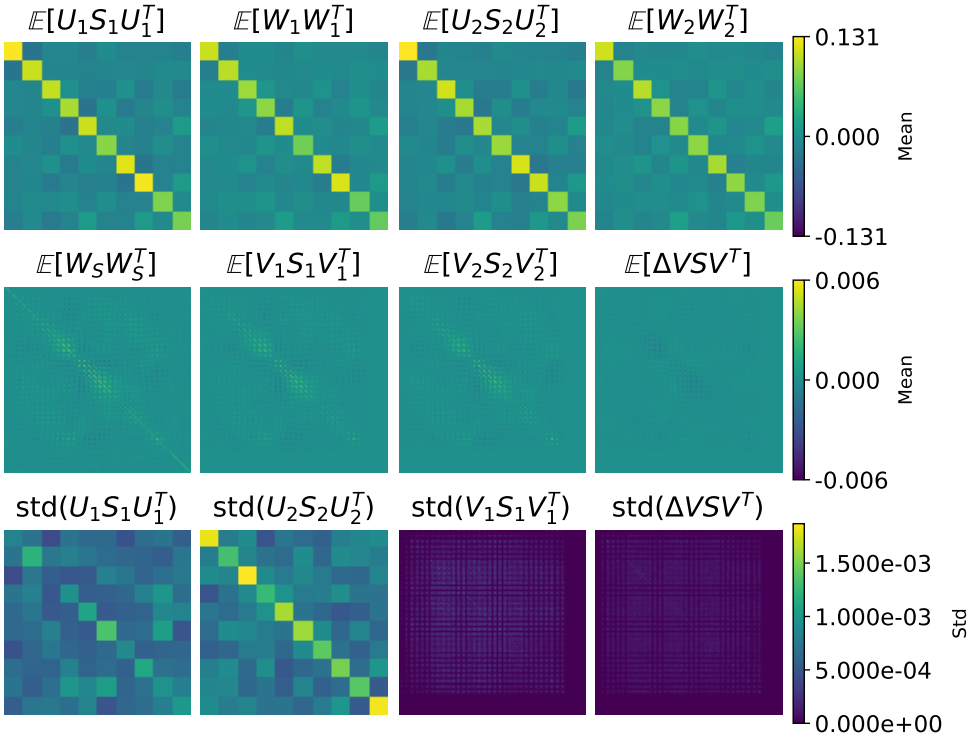

Figure 23: Representational similarity matrices (RSMs) of tasks target values (top) and inputs (bottom) compared to the RSM of the neural representations of a linear MTL NN trained on a batch (N=50000) of the Baseline Multi-MNIST dataset ($\eta = 0.01$, N_S=20) after convergence. Top row: (**A**) RSM_{1o} from the network at convergence compared to the RSMs of the target values of (**B**) task 1,(**C**) task 2 and (**D**) the combined task space ξ. Bottom row: RSM_I of the shared hidden representation of the network (**E**) compared to the RSMs of the input of (**F**) task 1, (**G**) task 2 and (**H**) the combined task space ξ.

Table 2: Alignment singular values α_i for the Permuted Multi-MNIST configuration, averaged over 3 seeds.

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
α_i	1.000	0.242	0.220	0.203	0.191	0.176	0.166	0.159	0.140	0.135
$\pm$ std	0.000	0.014	0.007	0.005	0.006	0.001	0.004	0.000	0.002	0.002

decreasing together, but Task 2 plateaus substantially earlier (around step 25000-30000) and at a lower loss. The gradient cosine similarity reflects this conflicting regime: after a weak initial rise, it turns negative and oscillates between roughly -0.2 and -0.55 for an extended period, before a slow partial recovery as gradient magnitudes vanish near convergence. The gradient magnitude curves show only a modest asymmetry: both tasks rise and fall over a similar timescale, with Task 2's peak arriving slightly earlier and marginally higher than Task 1's.

We further extend our RSM analysis to the Permuted Multi-MNIST setting. Because the tasks are only weakly aligned, joint training reshapes the representations far less than in the Baseline configuration, so the MTL and STL RSMs remain close ($\mathbf{\Delta USU}^\top, \mathbf{\Delta VSV}^\top \approx 0$). This is confirmed in Figure 24.

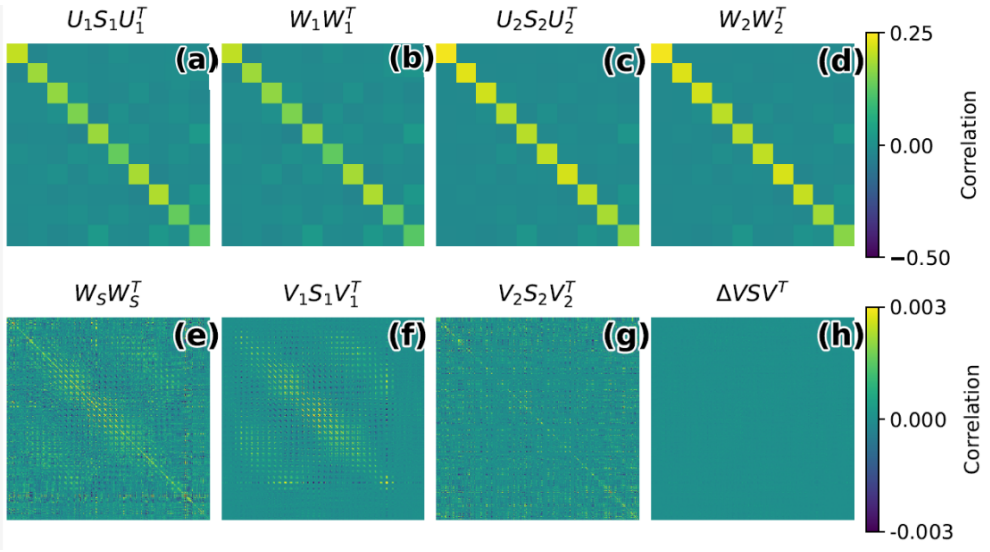

Figure 24: Representational similarity matrices (RSMs) of tasks target values (top) and inputs (bottom) compared to the RSM of the neural representations of a linear MTL NN trained on a batch (N=50000) of the permuted Multi-MNIST dataset ($\eta = 0.01$, N_S=20) after convergence. Top row: (**A**) RSM_{1_o} from the network at convergence compared to the RSMs of the target values of (**B**) task 1, (**C**) task 2 and (**D**) the combined task space ξ. Bottom row: RSM_I of the shared hidden representation of the network (**E**) compared to the RSMs of the input of (**F**) task 1, (**G**) task 2 and (**H**) the combined task space ξ.

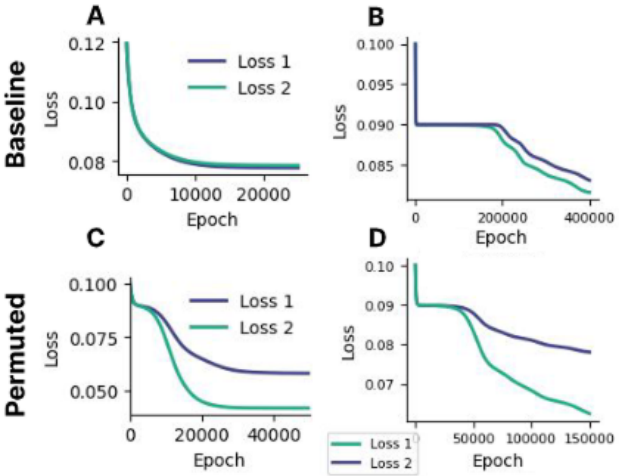

Figure 25: Comparison of training losses for Baseline (**A**)-(**B**) Multi-MNIST and Permuted (**C**)-(**D**) multi-MNIST on two-layer linear network (**A**) and (**C**) and a four-layer linear network (**B**)-(**D**). The increased depth does not impact the analyses of the learning curves as compared to the main text.

D.4 Teacher-Student MTL Regression Task

For this setup, we generalise the teacher-student task in Braun et al. (2022) to an MTL setup. We sample the input $X \sim \mathcal{N}(0, \sigma_X)$ with size $N_D \times N_i$ and whiten it such that $\frac{1}{N_D} XX^\top = \mathbf{I}$. To make sure we create tasks that are fully learnable with a linear neural network, we generate,

$$\mathbf{Y}_1 = \mathbf{W}_1(0)\mathbf{W}_S(0)\mathbf{X} \quad \mathbf{Y}_2 = \mathbf{W}_2(0)\mathbf{W}_S(0)\mathbf{X}, \tag{158}$$

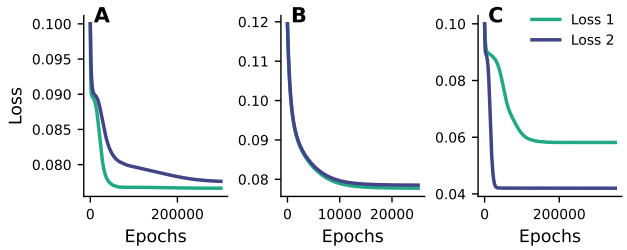

Figure 26: Comparison of training losses for Baseline multi-MNIST trained on a two-layer linear MTL network with $\mathbf{A} : \omega_2 = 0.25, \mathbf{B} : \omega_2 = 1, \mathbf{C} : \omega_2 = 0.75$

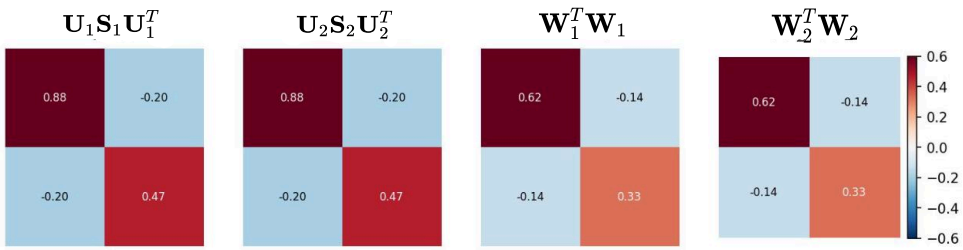

Figure 27: Comparison of RSMs of target values of aligned Teacher-Student regression tasks and the corresponding RSMs of internal layers obtained from a two-layer linear MTL with $N_1 = 10$, $N_{1o} = N_{2o} = 2$ and $N_S = 5$. The task-specific MTL neural RSM matches the theoretical prediction and is lower than the task-specific RSM expected when the task is trained in STL

where $\mathbf{W}_1 \sim \mathcal{N}(0, \sigma_{TS1})$ (size $N_1 \times N_s$), $\mathbf{W}_2 \sim \mathcal{N}(0, \sigma_{TS2})$ (size $N_2 \times N_s$) are randomly sampled from a gaussian distribution . Then $\mathbf{W}_S$ (size $N_s \times N_i$ is chosen such that the zero-balanced initialization condition is satisfied.

The tasks in equation (158) are learned by a neural network with input dim N_i, shared dim N_s, and task-specific layers N_1, N_2 with uniform weighting. Note that the scale of the initial weights of the neural network, σ_{NN} is not necessarily the same as σ_{TS}. The weights are also not necessarily the same. In Figure 27, we plot the RSMs of the internal layers of the neural network trained on such a task and compare this with our theoretical predictions from section 5.1.

D.5 Experiments on NYUv2

To analyse how our framework extends to more complex data, we conducted additional experiments on a higher-dimensional dataset with more than two tasks, i.e., NYUv2 Silberman et al. (2012) and discuss the resulting insights.

The NYUv2 dataset Silberman et al. (2012) is a widely used benchmark for indoor scene understanding. It contains RGB-D images captured with the RGB and depth sensors of a Microsoft Kinect, with 795 images used for training and 654 for validation. Following Lin & Zhang (2023), we use the preprocessed version provided by Liu et al. (2019), where each RGB image is resized to a tensor of size $3 \times 284 \times 284$ and annotated for three prediction tasks commonly used in multi-task learning benchmarks. Semantic segmentation assigns each pixel to one of 13 indoor object or scene categories. Depth estimation predicts the distance from the camera to the visible surfaces in the scene. Surface normal estimation predicts, for each pixel, the 3D orientation of the corresponding surface.

D.5.1 Measuring alignment in high-dimensional tasks

Since the definition of task alignment introduced in Section 2.2 is tied to task-specific input-output correlation matrices in the linear setting, applying it directly to high-dimensional dense prediction tasks such as NYUv2

would be hard to interpret. Instead, we construct an empirical surrogate that follows the same underlying rationale: for each task, we estimate a task-relevant subspace of the input, and then measure alignment through the overlap of these subspaces. In this sense, our NYUv2 alignment measure should be interpreted not as the exact quantity appearing in Section 2.2, but as an input-subspace analogue motivated by the same intuition.

Measuring alignment between depth and normal tasks. To empirically measure data-level task alignment on NYUv2, we adapt the theoretical overlap matrix $\mathbf{O}_{12}$ to the high-dimensional dense prediction setting as follows. Starting from the N=636 raw RGB training images $\mathbf{X} \in \mathbb{R}^{3\times 288\times 384}$, we flatten each image into a vector and apply PCA via the truncated SVD of the mean-centred data matrix $\mathbf{X} \in \mathbb{R}^{N\times p}$, where $p = 3 \times 288 \times 384 = 331{,}776$, retaining the top d' whitened principal components. This yields a reduced image-level representation $\mathbf{z}_n \in \mathbb{R}^{d'}$ for each image n, where each dimension has unit variance and the components are ordered by decreasing explained variance of the input.

Since the task targets are spatially dense: $\mathbf{y}_{\text{depth}} \in \mathbb{R}^{1\times 288\times 384}$ giving the per-pixel distance to the visible surface, and surface normals $\mathbf{y}_{\text{normal}} \in \mathbb{R}^{3\times 288\times 384}$ giving the per-pixel 3D surface orientation, we spatially downsample them by a factor of 16 via average pooling, yielding a grid of $H_s \times W_s = 18 \times 24 = 432$ spatial locations per image and $N \times 432 = 274{,}752$ paired samples in total. Each spatial location s in image n inherits the image-level representation $\mathbf{z}_n$ of its parent image, giving paired samples $(\mathbf{z}_n,\, \tilde{\mathbf{y}}_{i,n,s})$ for each task i, image n, and spatial location s, where $\tilde{\mathbf{y}}_{i,n,s} \in \mathbb{R}^{k_i}$ denotes the downsampled target of task $\mathcal{T}_i$ at location s in image n, with $k_{\text{depth}} = 1$ and $k_{\text{normal}} = 3$. We then compute the input-output correlation matrix

$$\mathbf{\Sigma}_{io} \;=\; \frac{1}{N \cdot P} \sum_{n=1}^{N} \sum_{s=1}^{P} \tilde{f}_{i,n,s} \tilde{\mathbf{z}}_n^{\top} \;\in\; \mathbb{R}^{k_i \times d'}, \tag{159}$$

where $P = 432$ is the number of spatial locations per image after downsampling, and tildes denote mean-centring over all (n, s) pairs. Taking the SVD $\mathbf{\Sigma}_{io} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^{\top}$ consistent with equation (8) yields right singular vectors $\mathbf{V}_i \in \mathbb{R}^{d' \times r}$, where $r = \min(k_i, d')$, spanning the directions in PCA-reduced input space most predictive of task $\mathcal{T}_i$. The overlap matrix $\mathbf{O}_{12} = \mathbf{V}_1^{\top} \mathbf{V}_2$ then directly instantiates Definition 2.1 and has a single singular value $\bar{\alpha} \in [0, 1]$ for the depth↔normal pair since $\min(k_{\text{depth}}, k_{\text{normal}}) = \min(1, 3) = 1$, which defines our alignment score. Statistical significance is assessed against a permutation baseline obtained by shuffling depth labels across images 20 times and recomputing $\bar{\alpha}$ each time, yielding a baseline of $\bar{\alpha}_{\text{perm}} \pm \sigma_{\text{perm}}$.

The segmentation task. In analogy to the depth and normal tasks, the segmentation labels were first downsampled by a factor of 16, resulting in a grid of dimensions 18×24. This downsampled segmentation mask was then converted into a one-hot encoding over the 13 segmentation classes, yielding a target of size $18 \times 24 \times 13$, and this was finally flattened into a single vector of length 5616 for each sample. The linear model was then trained with plain MSE on this downsampled, flattened one-hot segmentation target rather than with a full-resolution cross-entropy loss.

Sensitivity to PCA dimension. To assess the sensitivity of $\bar{\alpha}$ to the choice of d', we repeat the alignment computation across a range of values from $d' = 25$ to $d' = 325$, corresponding to 65.8% and 95.0% of explained input variance , respectively, with results reported in Table 4. At $d' = 25$, the observed alignment of $\bar{\alpha} = 0.295$ is indistinguishable from the permutation baseline (-0.1σ). The observed alignment peaks in magnitude at $d' = 50$ ($\bar{\alpha} = 0.386$, 2.8σ above baseline). Beyond $d' = 50$, the observed $\bar{\alpha}$ decreases monotonically while statistical significance increases monotonically, reaching 7.4σ at $d' = 325$. This pattern arises because adding higher-frequency components dilutes the observed overlap while simultaneously shrinking the permutation baseline: random unit vectors become increasingly orthogonal as the ambient dimension grows, so chance alignment falls faster than the observed signal. We therefore report $d' = 325$ as our primary setting, corresponding to the principled 95% variance threshold, which yields the most statistically robust estimate of $\bar{\alpha} = 0.296$ ($\bar{\alpha}_{\text{perm}} = 0.084 \pm 0.029$, 7.4σ above baseline).

Role of downsampling. Spatial downsampling serves two purposes: it renders the computation tractable, retaining all $288 \times 384 = 110{,}592$ pixels per image would require holding approximately 14GB of paired samples in memory, and it implicitly smooths the target maps, reducing the influence of high-frequency label

Table 3: Data-level task alignment $\bar{\alpha}$ for all three NYUv2 task pairs at $d' = 325$ (95% explained variance). The permutation baseline is estimated over 20 label shuffles. For the normals–segmentation pair, the individual singular values of $\mathbf{O}_{12}$ are also reported, since $\min(k_1, k_2) = 3 > 1$.

Task Pair	$\bar{\alpha}$	**Singular Values** α_i	**Baseline**	**Significance**
Depth ↔ Normals	0.296	$\{0.296\}$	0.084 ± 0.029	7.4σ
Depth ↔ Segmentation	0.543	$\{0.543\}$	0.211 ± 0.037	9.0σ
Normals ↔ Segmentation	0.536	$\{0.775,\ 0.696,\ 0.137\}$	0.206 ± 0.020	16.7σ

Table 4: Alignment $\bar{\alpha}$ between depth and surface normals as a function of the number of retained PCA components d'. The permutation baseline $\bar{\alpha}_{\text{perm}}$ reports the mean and standard deviation over 20 random label shuffles. Significance is expressed in standard deviations above the baseline.

d'	**Variance Explained**	$\bar{\alpha}$	$\bar{\alpha}_{\textbf{perm}} \pm \sigma$	**Significance**
25	0.658	0.295	0.302 ± 0.112	-0.1σ
50	0.744	0.386	0.208 ± 0.064	2.8σ
75	0.791	0.337	0.169 ± 0.077	2.2σ
100	0.824	0.331	0.150 ± 0.069	2.6σ
150	0.869	0.311	0.129 ± 0.056	3.3σ
200	0.901	0.294	0.110 ± 0.045	4.1σ
275	0.934	0.280	0.093 ± 0.041	4.6σ
325	0.950	0.296	0.084 ± 0.029	7.4σ

noise on the cross-covariance estimate while retaining the spatially coherent structure that is most relevant for measuring task alignment.

Alignment between the three NYUv2 task pairs. We measure data-level alignment between all three NYUv2 task pairs at $d' = 325$, corresponding to 95% of explained input variance, with results reported in Table 3. All three pairs exhibit statistically robust alignment well above the permutation baseline. The two highest-aligned pairs are depth ↔ segmentation ($\bar{\alpha} = 0.543$, 9.0σ above baseline) and normals ↔ segmentation ($\bar{\alpha} = 0.536$, 16.7σ above baseline), which are nearly indistinguishable in magnitude but differ in spectral structure: while depth ↔ segmentation produces a single singular value by construction (since $\min(k_{\text{depth}}, k_{\text{seg}}) = \min(1, 13) = 1$), the normals ↔ segmentation spectrum $\{0.775, 0.696, 0.137\}$ reveals that alignment is concentrated in two of the three normal modes, with the third largely orthogonal to the segmentation-predictive subspace. Depth ↔ normals is the least aligned pair by a substantial margin ($\bar{\alpha} = 0.296$, 7.4σ above baseline), suggesting that the input directions most predictive of scene depth occupy a largely distinct region of PCA-reduced input space from those most predictive of local surface orientation.

D.5.2 Negative Interference

Experiments with Two-Layer Linear Network We train each pair of tasks separately using full-batch gradient descent (batch size = 636, i.e., the full training set) with MSE loss and a learning rate of 10^{-5}, to stay close to our theoretical model. For each pair, the network is a two-layer linear multi-task model with a shared hidden representation of size $N_S = 16$, satisfying the no-bottleneck assumption (A3) for both tasks. The architecture consists of a shared linear input-to-hidden map followed by two task-specific linear readout heads. We normalize the depth target and initialize $\mathcal{N}(0, 10^{-4})$. We plot training curve until the gradient magnitudes remain close to zero for an extended period of time.

Predicting negative interference from the data. We stress that these NYUv2 results are illustrative rather than conclusive. Measuring task alignment on NYUv2 requires assumptions (e.g. the PCA reduction and dimension choices used to estimate $\mathbf{O}_{12}$) whose restrictiveness we cannot fully assess, and training two-layer linear networks on NYUv2 is computationally inefficient, making a systematic study across settings costly. Rather than introduce further assumptions, we present a small set of illustrative curves and leave a more thorough investigation to future work.

Figures 29-30show task losses, gradient cosine similarity (GradSim), and gradient magnitudes (GradMag) over training for each of the three NYUv2 task pairs. The segmentation-normals pair (Figure 30) shows an extended period of high, positive cosine similarity alongside a single, shared early peak in gradient magnitude for both tasks, before cosine similarity gradually declines and becomes mildly negative later in training. This is broadly consistent with the mixed aligned spectrum signature described in Appendix D.2, and plausibly reflects the two strongly aligned modes (α=0.775,0.696,0.137) identified in Table 3. Depth-normals (Figure 28) and segmentation-depth (Figure 29) instead show only a brief initial period of positive cosine similarity before dropping sharply negative, reaching minima of roughly -0.75 and -0.6 respectively, with partial recovery and further oscillation afterward. In both cases, the gradient magnitude plots show one task peaking substantially earlier and higher than the other, giving the peaks a more sequential than simultaneous appearance, rather than the single shared peak seen for segmentation-normals.

We note, however, that depth appears in both of these latter pairs, and the similarity in shape between depth-normals and segmentation-depth could reflect a property specific to the depth task or its loss scaling, rather than being purely a consequence of their alignment spectra. Disentangling these explanations would require additional task pairs and network architectures, which we leave for future work.

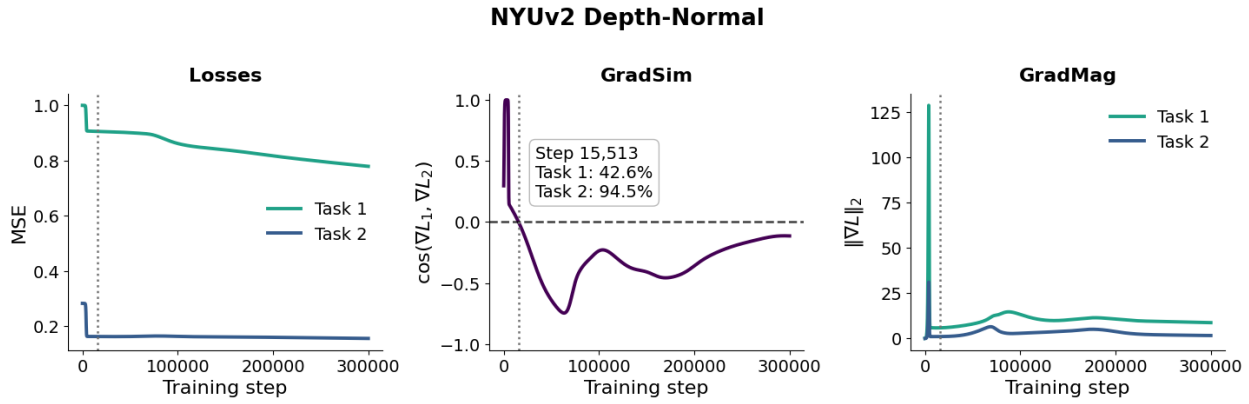

Figure 28: Losses, gradient similarity and gradient magnitudes of NYUv2 tasks Depth and Normal trained by a two-layer linear network. The percentages indicate the reduction in each task's loss relative to its value at training step 300k when the gradient cosine similarity first becomes negative.

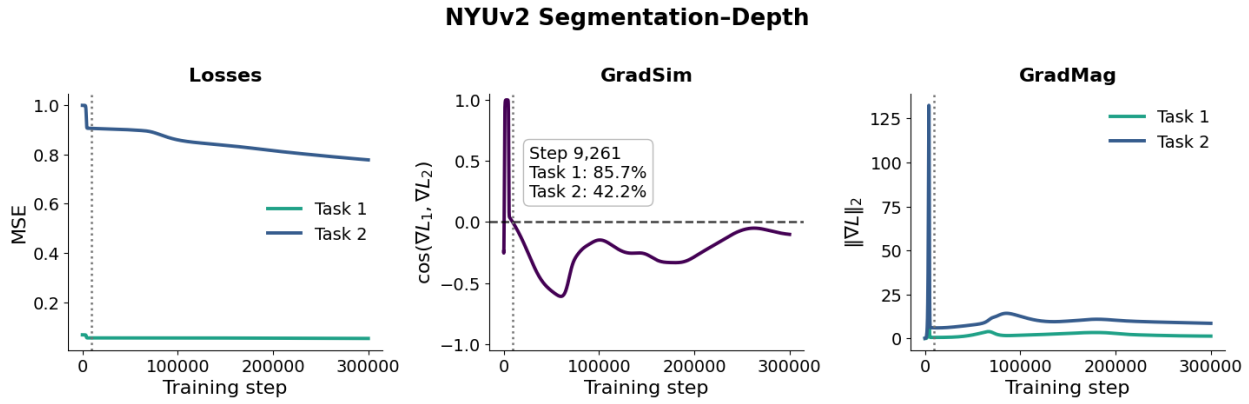

Figure 29: Losses, gradient similarity and gradient magnitudes of NYUv2 tasks Segmentation and Depth trained by a two-layer linear network. The percentages indicate the reduction in each task's loss relative to its value at training step 300k when the gradient cosine similarity first becomes negative.

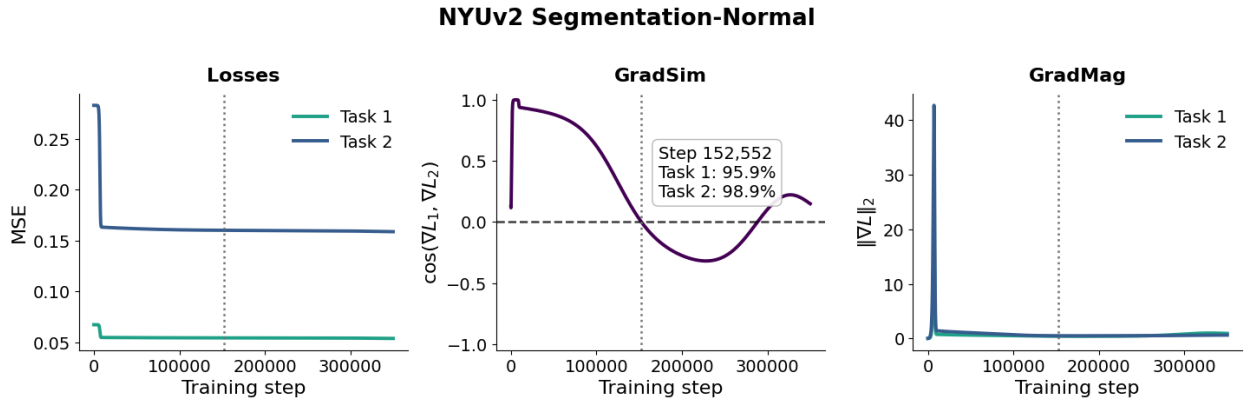

Figure 30: Losses, gradient cosine similarity, and gradient magnitudes for the NYUv2 Segmentation and Normal tasks trained with a two-layer linear network. The percentages indicate the reduction in each task's loss relative to its value at training step 300k when the gradient cosine similarity first becomes negative.

E Figures

In this section, we provide the experimental details for the figures in the main paper. We include a Python notebook for each figure to facilitate reproducibility.

Figure 1. Representation of a two-layer linear multi-task network created with Figma.

Figure 2. Schematic example of how the singular value decompositions of two tasks are combined into the "total MTL" space ξ in a non-trivial way, creating new modes which are related to the original tasks in complex ways. Figure created with Figma.

Figure 3. Comparison of the learning dynamics of Regression tasks with different degrees of alignment as predicted by the analytical solution (dotted lines) in Theorem 3.2 and numerical simulations (solid lines). The regression tasks have dimension $N_1 = 5, No_{1,2} = 1$ and are constructed following the procedure outlined in Appendix D.1 by setting $\gamma = 1$ for fully aligned, $\gamma = 0.3$ for fully conflicting and $\gamma = 0$ for fully orthogonal task pairs. To create the magnitude imbalance, $\alpha_2 = 5$, yielding $\Delta M = 0.2$. We plot the dynamics resulting from training a two-layer linear MTL network with $N_s = 2$, $N_D = 10$, $\eta = 0.0002$ and $\sigma_W = 0.00001$. Figure 21 reproduces Figure 3 across multiple seeds, showing the mean and standard deviation.

Figure 4. Comparison of RSMs for regression task pairs ($N_1 = 4$, $N_{o1} = N_{o2}$) with different alignment, generated by constructing by an orthonormal basis $\mathbf{R}$ and choosing B_1 as the first N_1 rows, $B_1^{\perp}$ as the remaining rows. As explained in Appendix D.1, the parameter γ determines the amount of alignment between the tasks while α_i determines the magnitude (im)balance. For the balanced tasks Figure 5(**A**) and 5(**C**), we set $\alpha_1 = \alpha_2 = 1$ while we change $\alpha_2 = 4$ for the plot in Figure 5(**B**). For the different degrees of alignment, we set $\gamma = 1$ for fully aligned, $\gamma = 0.3$ for fully conflicting and $\gamma = 0$ for fully orthogonal task pairs. Across 200 seeds, the output RSMs $\mathbf{U}_1\mathbf{S}_1\mathbf{U}_1^\top$, $\mathbf{U}_2\mathbf{S}_2\mathbf{U}_2^\top$, and $\mathbf{U}_\xi\mathbf{S}_\xi\mathbf{U}_\xi^\top$ have negligible variance (std $\approx 10^{-16}$). The single-task input RSMs $\mathbf{V}_i\mathbf{S}_i\mathbf{V}_i^\top$ vary across seeds (std $\approx 10^{-2}$, reflecting the random input basis), yet their combined difference $\mathbf{\Delta VSV}^\top$ is again invariant (std $\approx 10^{-16}$), confirming that $\mathbf{\Delta VSV}^\top = \mathbf{0}$ holds exactly.

Figure 5. Learning dynamics and representational similarity matrices (RSMs) for a two-layer multi-task network with linear (solid lines) and ReLU activations ($N_s = 20$, $\eta = 10^{-2}$, $\sigma_W = 10^{-4}$), trained on the Baseline (top) and Permuted (bottom) Multi-MNIST datasets (Appendix D.3). The dimensions of the dataset are; $N_1 = 1296; N_{o1} = N_{o2} = 10$ and $N_D = 50000$. The singular values of the overlap matrix for the Baseline pair of tasks are predominantly aligned across all directions, with no sharp drop in the spectrum $\alpha_i \in [1, 0.53]\}$. In contrast, we have $\alpha_i \in [1.000, 0.135]$ for the Permuted configuration. **Plots A-B:** Learning curves for two tasks, showing close to simultaneous learning in the Baseline setting (**A**) and a mixture of sequential and simultaneous learning in the Permuted setting (**B**). Alongside the losses, we also plot the cosine similarity between the task-specific gradients and the gradient magnitudes to establish a connection with common measures of negative interference. **Plots C-D:**($\eta = 10^{-2}$, $\sigma_W = 10^{-2}$) Comparison between single-task RSMs $\mathbf{U}_i\mathbf{S}_i\mathbf{U}_i^\top$ and the RSMs of the task-specific network weights $\mathbf{W}_i\mathbf{W}_i^\top$ at 200K epochs (zero gradient magnitude and flat losses), revealing weaker structural alignment relative to STL in the Baseline case (**C**) and close agreement with STL representations in the Permuted case (**D**) . Their differences $\Delta USUS$ are presented on subplots (**E**-**E**). Finally, note that the subplot labels and composition of the final Figure were created in Figma.

LLM Usage. LLMs were employed exclusively to refine the clarity of the writing and to enhance the code for improved visual presentation; the underlying data and analyses remain entirely unaffected.