Understanding Unimodal Bias in Multimodal Deep Linear Networks

Yedi Zhang [1] **Peter E. Latham** [1] **Andrew Saxe** [1 2]

Abstract

Using multiple input streams simultaneously to train multimodal neural networks is intuitively advantageous but practically challenging. A key challenge is unimodal bias, where a network overly relies on one modality and ignores others during joint training. We develop a theory of unimodal bias with multimodal deep linear networks to understand how architecture and data statistics influence this bias. This is the first work to calculate the duration of the unimodal phase in learning as a function of the depth at which modalities are fused within the network, dataset statistics, and initialization. We show that the deeper the layer at which fusion occurs, the longer the unimodal phase. A long unimodal phase can lead to a generalization deficit and permanent unimodal bias in the overparametrized regime. Our results, derived for multimodal linear networks, extend to nonlinear networks in certain settings. Taken together, this work illuminates pathologies of multimodal learning under joint training, showing that late and intermediate fusion architectures can give rise to long unimodal phases and permanent unimodal bias. Our code is available at: https://yedizhang.github.io/unimodal-bias.html.

1. Introduction

The success of multimodal deep learning hinges on effectively utilizing multiple modalities (Baltrušaitis et al., 2019; Liang et al., 2024). However, some multimodal networks overly rely on a faster-to-learn or easier-to-learn modality and ignore the others during joint training (Goyal et al., 2017; Cadene et al., 2019; Wang et al., 2020; Gat et al., 2021; Peng et al., 2022). For example, Visual Question Answering models should provide a correct answer by both "listening" to the question and "looking" at the image (Agrawal et al., 2016), whereas they tend to overly rely on the language modality and ignore the visual modality (Goyal et al., 2017; Agrawal et al., 2018; Hessel & Lee, 2020). This phenomenon has been observed in a variety of settings, and has several names: unimodal bias (Cadene et al., 2019), greedy learning (Wu et al., 2022), modality competition (Huang et al., 2022), modality laziness (Du et al., 2023), and modality underutilization (Makino et al., 2023). In this paper, we adopt the term unimodal bias to refer to the phenomenon in which a multimodal network learns from different input modalities at different times during joint training.

The extent to which multimodal networks exhibit unimodal bias depends on both the dataset and the multimodal network architecture. Regarding datasets, practitioners managed to alleviate the bias by building more balanced multimodal datasets (Goyal et al., 2017; Agrawal et al., 2018; Hudson & Manning, 2019). Regarding multimodal network architectures, empirical work has shown that unimodal bias emerges in jointly trained late fusion networks (Wang et al., 2020; Huang et al., 2022) and intermediate fusion networks (Wu et al., 2022), while early fusion networks may encourage usage of all input modalities (Gadzicki et al., 2020; Barnum et al., 2020).

Despite empirical evidence, there is scarce theoretical understanding of how unimodal bias is affected by the network configuration, dataset statistics, and initialization. To fill the gap, we study multimodal deep linear networks where unimodal bias manifests in ways consistent with those observed in complex networks, thereby serving as a prerequisite for theoretical understanding of complex networks. We show that unimodal bias is conspicuous in late and intermediate fusion linear networks but not in early fusion linear networks. Intermediate and late fusion linear networks learn one modality first and the other after a delay, yielding a phase in which the multimodal network implements a unimodal function. We calculate the duration of the unimodal phase in terms of parameters of the network and statistics of the dataset. We find that a deeper fusion layer within the multimodal network, stronger correlations between input modalities, and greater disparities in input-output correlations for each modality all prolong the unimodal phase. In the overparameterized regime, long unimodal phases creates a dilemma between overfitting one modality and underfitting the other, resulting in permanent unimodal bias and a

[1]Gatsby Computational Neuroscience Unit, University College London [2]Sainsbury Wellcome Centre, University College London. Correspondence to: Yedi Zhang <yedi@gatsby.ucl.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

generalization deficit. Our results apply to nonlinear networks in certain settings, providing insights for examining, diagnosing, and curing unimodal bias in a broader range of realistic cases.

In summary, our contributions are the following: (i) We provide a theoretical explanation for why unimodal bias is conspicuous in late and intermediate fusion linear networks but not in early fusion linear networks. (ii) We calculate the duration of the unimodal phase in mulitmodal learning with late and intermediate fusion linear networks, as a function of the network configuration, correlation matrices of the dataset, and initialization scale. (iii) We analyze the mis-attribution during the unimodal phase and the superficial modality preference. (iv) We reveal that long unimodal phases lead to a generalization deficit and permanent unimodal bias in the overparameterized regime. (v) We validate our findings with numerical simulations of multimodal deep linear networks and certain nonlinear networks.

1.1. Related Work

Several attempts have been made to understand unimodal bias. On the empirical front, Wang et al. (2020); Gat et al. (2021); Wu et al. (2022); Peng et al. (2022); Kleinman et al. (2023) have proposed metrics to assess unimodal bias across various multimodal models. Their metrics contributed to multimodal learning applications but did not lend a theory. On the theoretical front, Huang et al. (2021; 2022) proved that multimodal learning can outperform unimodal learning but may fail to deliver due to modality competition. In such failed cases, multiple modalities compete; only a subset of modalities correlated more with their encoding network's random initialization will win and be learned by the multimodal network. However, their study was restricted to two-layer late fusion networks and did not cover the influence of different networks and datasets upon the unimodal bias. Our work studies multimodal deep linear networks with three different fusion schemes, and we analytically reveal the relationship between unimodal bias, network configuration, and dataset statistics.

Our work builds on a rich line of theoretical analyses of deep linear networks. Fukumizu (1998); Saxe et al. (2014; 2019); Lampinen & Ganguli (2019); Braun et al. (2022); Shi et al. (2022); Atanasov et al. (2022) obtained exact solutions to the gradient descent dynamics of deep linear networks with whitened input. Gidel et al. (2019); Advani et al. (2020); Tarmoun et al. (2021) derived reductions and a special-case solution assuming spectral initialization for the same dynamics but with unwhitened input. Balancing properties in linear networks were discovered and proved in (Fukumizu, 1998; Arora et al., 2018; Du et al., 2018; Ji & Telgarsky, 2019). Nonetheless, previous solutions do not directly address our problem because multimodal deep linear networks are not fully connected and multimodal datasets generally do not have whitened input. We thus develop new analytic tools for multimodal deep linear networks, extending the theoretical landscape of linear networks.

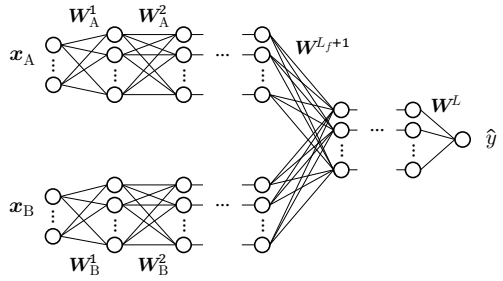


Figure 1. Schematic of a multimodal fusion network with total depth L and fusion layer at L_f.

2. Problem Setup

2.1. Multimodal Data

Let $\boldsymbol{x} \in \mathbb{R}^D$ represent an arbitrary multimodal input and $y \in \mathbb{R}$ be its scalar target output. We are given a dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^P$ consisting of P samples. For simplicity, we assume there are two modalities, A and B, with full input $\boldsymbol{x} = [\boldsymbol{x}_\mathrm{A}, \boldsymbol{x}_\mathrm{B}]$. Since we study multimodal linear networks with mean square error loss, the learning dynamics depends only on the correlation matrices of the data (Fukumizu, 1998; Saxe et al., 2014). We denote the input correlation matrix as $\boldsymbol{\Sigma}$ and input-output correlation matrix as $\boldsymbol{\Sigma}_{y\boldsymbol{x}}$; these are given explicitly by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_\mathrm{A} & \boldsymbol{\Sigma}_\mathrm{AB} \\ \boldsymbol{\Sigma}_\mathrm{BA} & \boldsymbol{\Sigma}_\mathrm{B} \end{bmatrix} = \begin{bmatrix} \left\langle \boldsymbol{x}_\mathrm{A}\boldsymbol{x}_\mathrm{A}^\top \right\rangle & \left\langle \boldsymbol{x}_\mathrm{A}\boldsymbol{x}_\mathrm{B}^\top \right\rangle \\ \left\langle \boldsymbol{x}_\mathrm{B}\boldsymbol{x}_\mathrm{A}^\top \right\rangle & \left\langle \boldsymbol{x}_\mathrm{B}\boldsymbol{x}_\mathrm{B}^\top \right\rangle \end{bmatrix}, \tag{1a}$$

$$\boldsymbol{\Sigma}_{y\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{A}} & \boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{B}} \end{bmatrix} = \begin{bmatrix} \left\langle y\boldsymbol{x}_\mathrm{A}^\top \right\rangle & \left\langle y\boldsymbol{x}_\mathrm{B}^\top \right\rangle \end{bmatrix}, \tag{1b}$$

where $\langle\cdot\rangle$ denotes the average over the dataset. We assume data points are centered to have zero mean, $\langle \boldsymbol{x} \rangle = \boldsymbol{0}$, and the input correlation matrix $\boldsymbol{\Sigma}$ has full rank, but make no further assumptions about the dataset.

2.2. Multimodal Fusion Linear Network

We study a multimodal deep linear network with total depth L and fusion layer at L_f defined as[1]

$$\begin{aligned} \hat{y}(\boldsymbol{x}; \boldsymbol{W}) &= \prod_{j=L_f+1}^{L} \boldsymbol{W}^j \left(\prod_{i=1}^{L_f} \boldsymbol{W}_\mathrm{A}^i \boldsymbol{x}_\mathrm{A} + \prod_{i=1}^{L_f} \boldsymbol{W}_\mathrm{B}^i \boldsymbol{x}_\mathrm{B} \right) \\ &\equiv \boldsymbol{W}_\mathrm{A}^\mathrm{tot} \boldsymbol{x}_\mathrm{A} + \boldsymbol{W}_\mathrm{B}^\mathrm{tot} \boldsymbol{x}_\mathrm{B} \\ &\equiv \boldsymbol{W}^\mathrm{tot} \boldsymbol{x} \,. \end{aligned} \tag{2}$$

[1]We abuse the notation $\prod_j \boldsymbol{W}^j$ to represent the ordered product of matrices with the largest index on the left and smallest on the right.

The overall network input-output map is denoted $\boldsymbol{W}^{\text{tot}}$, and the map for each modality is denoted $\boldsymbol{W}_{\text{A}}^{\text{tot}}, \boldsymbol{W}_{\text{B}}^{\text{tot}}$. We use $\boldsymbol{W}$ to denote all weight parameters collectively. We assume the number of neurons in both pre-fusion layer branches is of the same order. A schematic of this network is given in Figure 1.

Our network definition incorporates bimodal deep linear networks of three common fusion schemes. Categorized by the multimodal deep learning community (Ramachandram & Taylor, 2017), the case where $L_f = 1$ is early or data-level fusion; $1 < L_f < L$ is intermediate fusion; and $L_f = L$ is late or decision-level fusion.

2.3. Gradient Descent Dynamics

The network is trained by optimizing the mean square error, $\mathcal{L} = \frac{1}{2P}\sum_{\mu=1}^{P}(y^\mu - \hat{y}^\mu)^2$, with gradient descent. In the limit of small learning rate, the gradient descent dynamics are well approximated by the continuous time differential equations; see Appendix B. For pre-fusion layers $1 \leq l \leq L_f$,

$$\tau \dot{\boldsymbol{W}}_{\text{A}}^{l} = \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^j \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\text{A}}^i\right)^\top \boldsymbol{e}_{\text{A}} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\text{A}}^i\right)^\top, \tag{3a}$$

$$\tau \dot{\boldsymbol{W}}_{\text{B}}^{l} = \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^j \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\text{B}}^i\right)^\top \boldsymbol{e}_{\text{B}} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\text{B}}^i\right)^\top. \tag{3b}$$

For post-fusion layers $L_f + 1 \leq l \leq L$,

$$\begin{aligned}\tau \dot{\boldsymbol{W}}^{l} = &\left(\prod_{j=l+1}^{L} \boldsymbol{W}^j\right)^\top \boldsymbol{e}_{\text{A}} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^j \prod_{i=1}^{L_f} \boldsymbol{W}_{\text{A}}^i\right)^\top \\ &+ \left(\prod_{j=l+1}^{L} \boldsymbol{W}^j\right)^\top \boldsymbol{e}_{\text{B}} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^j \prod_{i=1}^{L_f} \boldsymbol{W}_{\text{B}}^i\right)^\top,\end{aligned} \tag{4}$$

where the time constant τ is the inverse of learning rate and $\boldsymbol{e}_{\text{A}}$ and $\boldsymbol{e}_{\text{B}}$ represent the correlation between the output error, $y - \hat{y}$, and the inputs, $\boldsymbol{x}_{\text{A}}, \boldsymbol{x}_{\text{B}}$,

$$\boldsymbol{e}_{\text{A}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}} - \boldsymbol{W}_{\text{A}}^{\text{tot}}\boldsymbol{\Sigma}_{\text{A}} - \boldsymbol{W}_{\text{B}}^{\text{tot}}\boldsymbol{\Sigma}_{\text{BA}}, \tag{5a}$$

$$\boldsymbol{e}_{\text{B}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}} - \boldsymbol{W}_{\text{A}}^{\text{tot}}\boldsymbol{\Sigma}_{\text{AB}} - \boldsymbol{W}_{\text{B}}^{\text{tot}}\boldsymbol{\Sigma}_{\text{B}}. \tag{5b}$$

The network is initialized with small random weights.

3. Two-Layer Multimodal Linear Networks

We first study two-layer multimodal linear networks, for which $L = 2$. There are two possible fusion schemes for two-layer networks according to our setup: early fusion, $L_f = 1$, as in Figure 2a and late fusion, $L_f = 2$, as in Figure 2d.

We explain that unimodal bias is conspicuous in late fusion linear networks but not in early fusion linear networks by analyzing their loss landscape in Section 3.1. We then focus on late fusion networks: Section 3.2 calculates duration of the unimodal phase; Section 3.3 specifies mis-attribution in the unimodal phase; Section 3.4 reveals the superficial modality preference; Section 3.5 demonstrates that a long unimodal phase harms generalization in the overparametrized regime.

3.1. Loss Landscape

As shown in Figures 2b and 2e, early fusion networks learn from both modalities almost simultaneously while late fusion networks learn two modalities at two separate times with a conspicuous unimodal phase in between. For both networks, the loss trajectories exhibit quasi-stage-like behaviors. As studied by Saxe et al. (2014; 2019); Jacot et al. (2021); Pesme & Flammarion (2023), linear networks trained from small initialization learn slowly for most of the time and move rapidly from one fixed point or saddle to the next with a sigmoidal transition stage. We show that early fusion networks have two manifolds of fixed points, corresponding to their one transition stage. In contrast, late fusion networks have two manifolds of fixed points and two manifolds of saddles, accounting for their two transition stages.

Early Fusion. There are two manifolds of fixed points in early fusion networks (Appendix C): one is an unstable fixed point at zero $\mathcal{M}_0$ and the other is a manifold of stable fixed points at the global pseudo-inverse solution $\mathcal{M}_*$,

$$\mathcal{M}_0 = \{\boldsymbol{W} | \boldsymbol{W} = \boldsymbol{0}\}; \tag{6a}$$

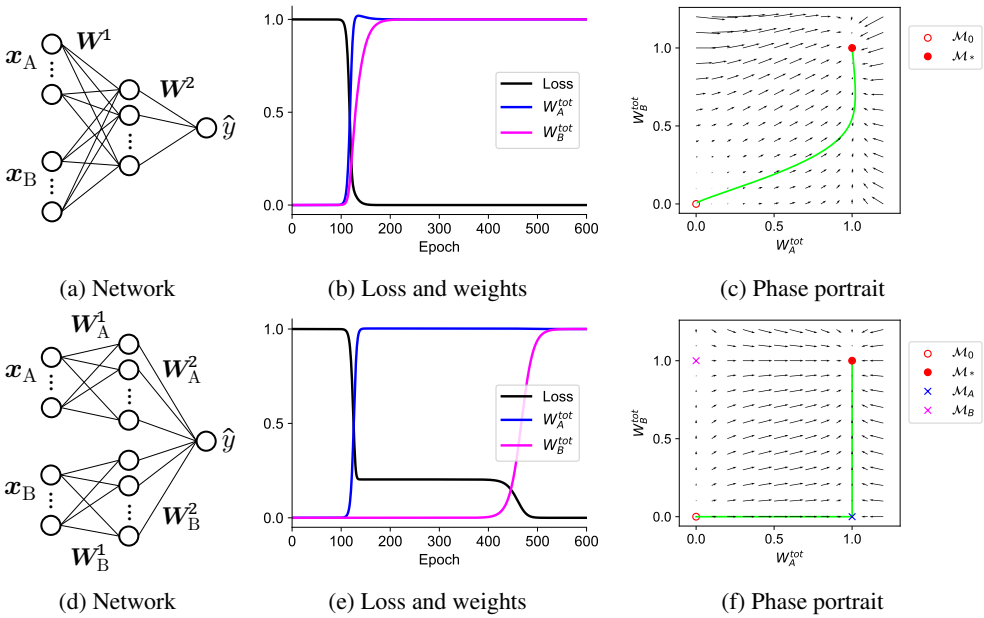
$$\mathcal{M}_* = \left\{\boldsymbol{W} | \boldsymbol{W}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}\right\}. \tag{6b}$$

The network starts from small initialization, which is close to the unstable fixed point $\mathcal{M}_0$. When learning progresses, the network escapes from the unstable fixed point $\mathcal{M}_0$ and converges to the global pseudo-inverse solution at $\mathcal{M}_*$ with one brief transition. We visualize the fixed points and the learning trajectory in the phase portrait in Figure 2c. Since there can only be one brief transition stage, all modalities are learned almost simultaneously in early fusion networks.

Late Fusion. Late fusion linear networks have the same two manifolds of fixed points $\mathcal{M}_0, \mathcal{M}_*$ as early fusion networks. In addition, late fusion linear networks have two manifolds of saddles $\mathcal{M}_{\text{A}}, \mathcal{M}_{\text{B}}$ (Appendix D.1), corresponding to learning one modality but not the other,

$$\mathcal{M}_{\text{A}} = \left\{\boldsymbol{W} | \boldsymbol{W}_{\text{A}}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}}{\boldsymbol{\Sigma}_{\text{A}}}^{-1}, \boldsymbol{W}_{\text{B}}^{\text{tot}} = \boldsymbol{0}\right\}; \tag{7a}$$

$$\mathcal{M}_{\text{B}} = \left\{\boldsymbol{W} | \boldsymbol{W}_{\text{A}}^{\text{tot}} = \boldsymbol{0}, \boldsymbol{W}_{\text{B}}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}}{\boldsymbol{\Sigma}_{\text{B}}}^{-1}\right\}. \tag{7b}$$



(a) Network (b) Loss and weights (c) Phase portrait

(d) Network (e) Loss and weights (f) Phase portrait

Figure 2. Effect of fusion point on learning dynamics and loss landscape. Top row: Early fusion. Bottom row: Late fusion. Both networks are trained with the same dataset. (a,d) Network schematic. (b,e) Training trajectories. (c,f) Phase portrait. Late fusion introduces two manifolds of saddles (blue and magenta crosses) into the loss landscape, causing learning trajectories to plateau near a unimodal solution. Experimental details are provided in Appendix J.

The late fusion linear network therefore undergoes two transition stages because the network first arrives and lingers near a saddle in $\mathcal{M}_{\mathrm{A}}$ or $\mathcal{M}_{\mathrm{B}}$ and subsequently converges to the global pseudo-inverse solution in $\mathcal{M}_*$. Visiting the saddle gives rise to the plateau in the loss that separates the time when the two modalities are learned. During the plateau, the network is unimodal. We visualize the fixed points and the learning trajectory in the phase portrait in Figure 2f.

3.2. Duration of the Unimodal Phase

We now quantify the duration of the unimodal phase. For definiteness, we designate the modality learned first to be modality A. We define t_{A} to be the time when the total weight of modality A reaches half of its associated plateau and similarly for t_{B}, as illustrated in Figures 3a and 3b. Because of the small random initialization, we can assume that the Frobenius norms of $\boldsymbol{W}_{\mathrm{A}}^1$ and $\boldsymbol{W}_{\mathrm{B}}^1$ are approximately equal at initialization, denoted u_0. We compute the times t_{A} and t_{B} in Appendix D.3 in leading order of the initialization u_0 and obtain

$$t_{\mathrm{A}} = \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} \ln \frac{1}{u_0}, \tag{8a}$$

$$t_{\mathrm{B}} = t_{\mathrm{A}} + \tau \frac{1 - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} {\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1} \boldsymbol{\Sigma}_{\mathrm{AB}}\right\|} \ln \frac{1}{u_0}. \tag{8b}$$

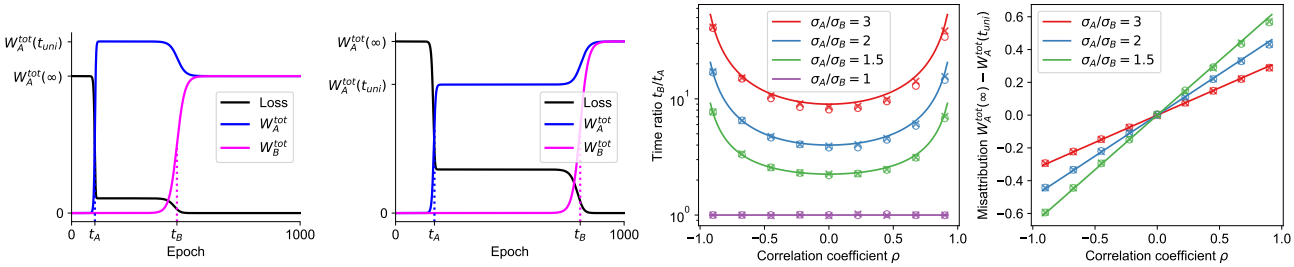
To compare the unimodal phase duration across different settings, we focus on the time ratio[2],

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} = 1 + \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} {\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1} \boldsymbol{\Sigma}_{\mathrm{AB}}\right\|}. \tag{9}$$

We note that the time ratio reduces to $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| / \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|$ if the cross correlation is zero, i.e., $\boldsymbol{\Sigma}_{\mathrm{AB}} = \boldsymbol{0}$. This accords with the intuition that $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|$ governs the speed at which modality A is learned and $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|$ governs the speed at which modality B is learned. When the cross correlation is nonzero, having learned modality A affects the speed at which modality B is learned during the unimodal phase. Specifically, the speed of modality B is reduced by $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} {\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1} \boldsymbol{\Sigma}_{\mathrm{AB}}$; see Appendix D.3.

We validate Equation (9) with numerical simulations in Figure 3c. With theoretical and experimental evidence, we conclude that stronger correlations between input modalities and a greater disparity in input-output correlations for each modality make the time ratio larger, indicating a longer unimodal phase. In the extreme case of maximum correlation, $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ are co-linear, and so one of them is redundant. In this case, the denominator $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} {\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1} \boldsymbol{\Sigma}_{\mathrm{AB}}$ is 0, and the ratio $t_{\mathrm{B}}/t_{\mathrm{A}}$ is ∞. Here later becomes never — the network learns to fit the output only with modality A and modality B will never be learned, as shown in Figure 8.

[2]We use $\|\cdot\|$ to notate the L2 norm of a vector or the Frobenius norm of a matrix in this paper.



(a) $\rho = 0.5$ loss and weights (b) $\rho = -0.5$ loss and weights (c) Time ratio (d) Mis-attribution

Figure 3. Duration of unimodal phase and amount of mis-attribution in two-layer late fusion linear networks. We consider scalar inputs $\boldsymbol{x}_{\mathrm{A}}, \boldsymbol{x}_{\mathrm{B}} \in \mathbb{R}$ with input covariance matrix parameterized as $\boldsymbol{\Sigma} = \left[\sigma_{\mathrm{A}}^2, \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}; \rho\sigma_{\mathrm{B}}\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}^2\right]$. The target output is generated as $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$. (a) Loss and total weight trajectories in two-layer late fusion networks when modalities are positively correlated. (b) Same as panel a but for negative correlations. (c) Time ratio $t_{\mathrm{B}}/t_{\mathrm{A}}$ as in Equation (9). (d) Amount of mis-attribution. In panel c and d, lines are theoretical predictions; circles are simulations of two-layer late fusion linear networks; crosses are simulations of two-layer late fusion ReLU networks. Experimental details are provided in Appendix J.

3.3. Mis-attribution in the Unimodal Phase

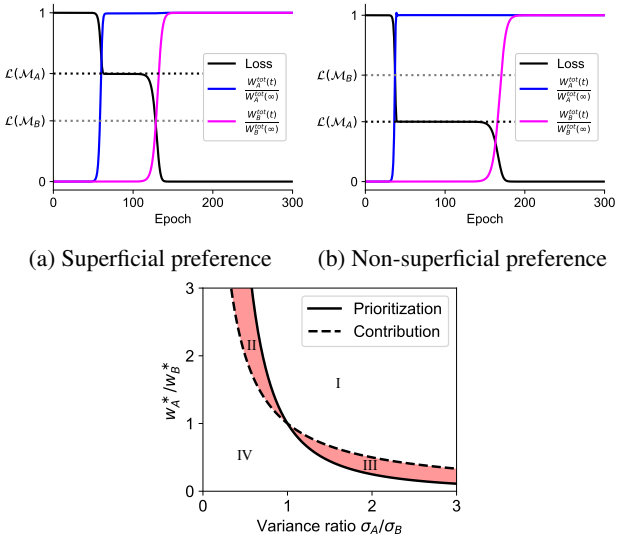
We take a closer look at the unimodal mode by highlighting a phenomenon we call mis-attribution. During the unimodal phase, $\boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}}$ fits the output as much as it can and the network mis-attributes some of the output contributed by modality B to modality A by exploiting their correlations. When modalities are correlated, the local pseudo-inverse solution differs from the global pseudo-inverse solution, i.e., $\left[\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1}, \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1}\right] \neq \boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}$. Specifically, the weights of modality A overshoot if modalities have a positive correlation as in Figure 3a and undershoot if negative as in Figure 3b. This mis-attribution is then corrected when modality B catches up at time t_{B}, and the network eventually converges to the global pseudo-inverse solution. In Figure 3d we demonstrate, using scalar input for clarity, that mis-attribution is more severe when modalities have stronger correlation.

When modalities are uncorrelated, late fusion networks do not mis-attribute during the unimodal phase because the local pseudo-inverse solutions are the same as the global pseudo-inverse solution. Weights for modality A converge to the global solution at time t_{A} and do not change thereafter, as in Figure 2e. In this case, the late fusion network behaves the same as two separately trained unimodal networks.

3.4. Superficial Modality Preference

We now look into which modality is learned first. Late fusion linear networks have what we call "superficial modality preference". They prioritize the modality that is faster to learn, which is not necessarily the modality that yields the larger decrease in loss.

From the time ratio expression given in Equation (9) and details in Appendix D.3.1, we see that which modality is learned first depends solely on the relative size of $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|$



(a) Superficial preference (b) Non-superficial preference

(c) Superficial region

Figure 4. Demonstration of superficial modality preference. A two-layer late fusion linear network is trained with two different dataset. (a,b) In both examples, modality A is learned first. The dotted black line marks the loss when the network visits $\mathcal{M}_{\mathrm{A}}$. The dotted gray line marks the loss if the network had instead visited $\mathcal{M}_{\mathrm{B}}$. (a) The prioritized modality is not the modality that contributes more to the output. (b) The prioritized modality is the modality that contributes more to the output. (c) Boundaries of which modality is prioritized and which modality contributes more to the output in terms of dataset statistics. In region I and III, modality A is learned first. In region I and II, modality A contributes more to the output. Thus in region II and III (shaded red), prioritization and contribution disagree, resulting in superficial modality preference. Experimental details are provided in Appendix J.

and $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{B}}\|$. Thus late fusion networks first learn the modality that has a higher correlation with the output, even though it may not be the modality that makes a larger contribution to the output. Under the following two conditions on dataset statistics,

$$\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{A}}\| > \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{B}}\|, \tag{10a}$$

$$\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{A}}\boldsymbol{\Sigma}_\mathrm{A}{}^{-1}\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{A}}^\top < \boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{B}}\boldsymbol{\Sigma}_\mathrm{B}{}^{-1}\boldsymbol{\Sigma}_{y\boldsymbol{x}_\mathrm{B}}^\top, \tag{10b}$$

modality A is faster to learn but modality B contributes more to the output. We present an example where these conditions hold in Figure 4a and where they do not in Figure 4b.

If $\boldsymbol{x}_\mathrm{A}, \boldsymbol{x}_\mathrm{B}$ are scalars and uncorrelated, the two inequality conditions in Equation (10) reduce to

$$\frac{\sigma_\mathrm{B}^2}{\sigma_\mathrm{A}^2} < \frac{w_\mathrm{A}^*}{w_\mathrm{B}^*} < \frac{\sigma_\mathrm{B}}{\sigma_\mathrm{A}}, \tag{11}$$

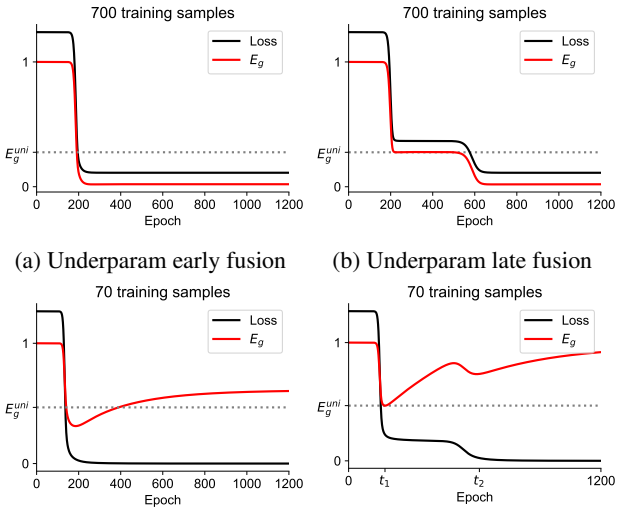
where $\sigma_\mathrm{A}, \sigma_\mathrm{B}$ are variances of $\boldsymbol{x}_\mathrm{A}, \boldsymbol{x}_\mathrm{B}$ and we assume the target output is generated as $y = w_\mathrm{A}^*\boldsymbol{x}_\mathrm{A} + w_\mathrm{B}^*\boldsymbol{x}_\mathrm{B}$. We plot the two conditions in Figure 4c. Region III satisfies the conditions we give in Equation (11). Region II corresponds to Equation (11) with flipped inequality signs, meaning the other superficial modality preference case where modality B is prioritized but modality A contributes more to the output. Hence, region II and III (shaded red) cover the dataset statistics where prioritization and contribution disagree and late fusion linear networks would prioritize learning the modality that contributes less to the output.

We note that the networks eventually converge to zero loss, regardless of which modality is learned first and how long the unimodal phase is. But they affect the generalization error in the overparameterized regime, which we discuss in the following section.

3.5. Underparameterization and Overparameterization

In the underparameterized regime, training loss closely tracks the corresponding generalization error as shown in Figures 5a and 5b because the training data is sufficient to accurately estimate the true data distribution. Analysis on the training loss applies to the generalization error as well. Both early and late fusion networks achieve a lower generalization error at convergence than that of a unimodal network (dotted gray line).

In the overparameterized regime, the number of samples is insufficient compared to the number of effective parameters, which is the input dimension for linear networks (Advani et al., 2020). As shown in Figure 5c, the overparameterized early fusion linear network learns both modalities during one transition stage. The generalization error decreases during the transition stage and increases afterwards, as predicted by theory (Le Cun et al., 1991; Krogh & Hertz, 1992; Advani et al., 2020). If early stopping is adopted, we obtain



(a) Underparam early fusion (b) Underparam late fusion

(c) Overparam early fusion (d) Overparam late fusion

Figure 5. Overparameterized and underparameterized two-layer early and late fusion linear networks. Inputs are 50-dimensional, i.e., $\boldsymbol{x}_\mathrm{A}, \boldsymbol{x}_\mathrm{B} \in \mathbb{R}^{50}$. (a) Loss and generalization error trajectories of a two-layer early fusion linear network trained with 700 examples. (b) Same as panel a but with late fusion. (c) Loss and generalization error trajectories of a two-layer early fusion linear network trained with 70 examples. (d) Same as panel c but with late fusion. The dotted gray line marks the lowest generalization error that a unimodal network could achieve with the same dataset. Experimental details are given in Appendix J.

a model that has learned from both modalities and does not substantially overfit. This model achieves a lower generalization error than its unimodal counterpart.

As shown in Figure 5d, the overparameterized late fusion linear network learns the faster-to-learn modality first and overfits this modality during the unimodal phase when the training loss plateaus but the generalization error increases. In this case, there is a dilemma between overfitting the first modality and underfitting the second. If optimal early stopping on the generalization error is adopted, training terminates at time t_1. The model at time t_1 has a strong and permanent unimodal bias because the first modality has just been learned while the second modality has not. If training terminates at time t_2, after both modalities are learned, the network has overfit the first modality. This can result in a generalization error worse than that in early stopping or what could be achieved by its unimodal counterpart. Thus overfitting is a mechanism that can convert the transient unimodal phase to a generalization deficit and permanent unimodal bias.

4. Deep Multimodal Linear Networks

We now consider the more general case of multimodal deep linear networks and examine how the fusion layer depth L_f affects the extent of unimodal bias.

4.1. Loss Landscape

Early Fusion. Similar to two-layer early fusion networks, deep early fusion networks learn from both modalities almost simultaneously, as shown in Figure 6a (purple curve). Saxe et al. (2014; 2019); Advani et al. (2020) have shown that for deep linear networks, depth slows down learning but does not qualitatively change the dynamics compared to two-layer linear networks. The weights associated with all input modalities escape from the initial zero fixed point $\mathcal{M}_0$ and converge to the pseudo-inverse solution fixed point in $\mathcal{M}_*$ in one transition stage.

Intermediate and Late Fusion. Deep intermediate and late fusion linear networks learn the two modalities with two separate transitions, as shown in Figure 6a; this is similar to what happens in two-layer late fusion networks. Due to the common terms in Equations (3) and (4) that govern the weight dynamics, the two manifolds of fixed points, $\mathcal{M}_0, \mathcal{M}_*$, and the two manifolds of saddles, $\mathcal{M}_{\mathrm{A}}, \mathcal{M}_{\mathrm{B}}$, exist for any $2 \leq L_f \leq L, L \geq 2$, comprising intermediate and late fusion linear networks of any configuration.

In what follows, we stick to the convention that modality A is learned first. Deep intermediate and late fusion networks start from small initialization, which is close to the zero fixed point $\mathcal{M}_0$. When modality A is learned in the first transition stage at time t_{A}, the network visits the saddle in $\mathcal{M}_{\mathrm{A}}$. After a unimodal phase, the network goes through the second transition at time t_{B} to reach the global pseudo-inverse solution fixed point in $\mathcal{M}_*$. Because the network is in the same manifold $\mathcal{M}_{\mathrm{A}}$ during the unimodal phase, our results in Section 3.3 on the mis-attribution in the unimodal phase and Section 3.4 on superficial modality preference in two-layer late fusion networks directly carry over to deep intermediate and late fusion linear networks.

As shown in Figure 6, the loss trajectories of networks with different $L_f \geq 2$ traverse the same plateau but they stay in the plateau for different durations. We thus quantify how the total depth L and fusion layer depth L_f affect the duration of the unimodal phase in Section 4.2.

4.2. Duration of the Unimodal Phase

We now calculate the duration of the unimodal phase in deep intermediate and late fusion linear networks, incorporating the new parameters L and L_f. The input-output correlation ratio is denoted $k = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\| / \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| \in (0, 1)$. We derive the time ratio through leading order approximation of the initialization scale in Appendix E. For $2 < L_f \leq L$, the time ratio is

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} = 1 + \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|} \times \frac{u_0^{L-L_f}}{(L_f-2)\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1}\|^{1-\frac{L_f}{L}}} I(L, L_f)^{-1}, \tag{12}$$

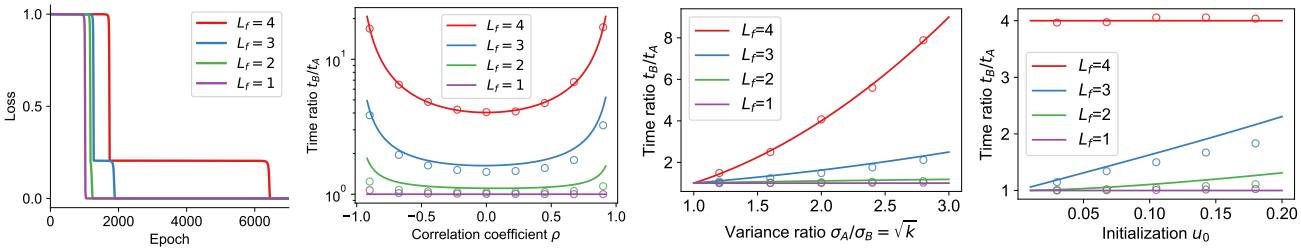
where

$$I(L, L_f) = \int_1^\infty \frac{\left[1 + \left(k + (1-k)\, x^{L_f-2}\right)^{\frac{2}{2-L_f}}\right]^{\frac{L_f-L}{2}}}{x^{L-1}} dx\,. \tag{13}$$

For $L_f = 2$, the expression is slightly different but qualitatively similar; see Appendix E.3. As shown in Figure 6, the theoretical prediction captures the trend that a deeper fusion layer L_f, a larger input-output correlation ratio k, and stronger correlations $\boldsymbol{\Sigma}_{\mathrm{AB}}$ between input modalities all prolong the duration of the unimodal phase. The qualitative influence of dataset statistics on the time ratio in deep intermediate and late fusion networks is consistent with what we have seen in two-layer late fusion networks.

We now look into the influence of the fusion layer depth. By setting $L_f = L$ in Equation (12), we find that the time ratio in deep late fusion networks reduces to the same expression as the two-layer late fusion case in Equation (9), which only involves dataset statistics but not the depth of the network or the initialization, since depth slows down the learning of both modalities by the same factor. In intermediate fusion linear networks, the time ratio is smaller than in late fusion networks, with a smaller ratio for a shallower fusion layer. In intermediate fusion linear networks, learning one modality changes the weights in its associated pre-fusion layers and the shared post-fusion layers. At time t_{A}, the pre-fusion layer weights of modality A and the shared post-fusion layer weights have escaped from the zero fixed point and grown in scale while the pre-fusion layer weights of modality B have not. During the unimodal phase, the shared post-fusion layer weights and the correlation between modality B and the output together drive the pre-fusion layer weights of modality B to escape from the zero fixed point. Thus having more shared post-fusion layers makes learning one modality more helpful for learning the other, shortening the unimodal phase. In essence, an early fusion point allows the weaker modality to benefit from the stronger modality's learning in the post-fusion layers.

We also note that the initialization scale affects the time ratio in intermediate fusion networks, as demonstrated in Figure 6d. Even amongst cases that all fall into the rich feature learning regime, the initialization scale has an effect



(a) Loss (b) Correlation sweep (c) Variance ratio sweep (d) Initialization sweep

Figure 6. Duration of unimodal phase in multimodal deep linear networks. Four-layer linear networks with fusion layer at $L_f = 1, 2, 3, 4$ are trained to fit $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$. The input covariance matrix is parameterized as $\boldsymbol{\Sigma} = \left[\sigma_{\mathrm{A}}^2, \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}; \rho\sigma_{\mathrm{B}}\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}^2\right]$. (a) An example of loss trajectories when input covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(4, 1)$. (b) Correlation coefficient sweep with $\sigma_{\mathrm{A}}/\sigma_{\mathrm{B}} = 2$ and initialization scale $u_0 = 0.1$. (c) Variance ratio sweep with $\rho = 0, u_0 = 0.1$. Note that $\sigma_{\mathrm{A}}/\sigma_{\mathrm{B}} = \sqrt{k}$ when $\rho = 0$. (d) Initialization scale sweep with $\sigma_{\mathrm{A}}/\sigma_{\mathrm{B}} = 2, \rho = 0$. In panel b to d, lines are theory; circles are simulations of four-layer linear networks with different fusion layer depth. Experimental details are provided in Appendix J.

on the time ratio, with a larger time ratio for a larger initialization scale. In Figure 6d, the simulations (circles) slightly deviate from theoretical predictions (lines) because our theoretical prediction is derived with small initialization and is thus less accurate for larger initialization. Nonetheless, the monotonic trend is well captured.

In summary, a deeper fusion layer, a larger input-output correlation ratio, stronger correlations between input modalities, and sometimes a larger initialization scale all prolong the unimodal phase in the joint training of multimodal deep linear networks with small initialization.

5. Discussion

To understand unimodal bias in multimodal learning, we have studied multimodal deep linear networks. In these networks, the unimodal bias already manifests itself in many ways consistent with those seen in complex networks. We now discuss how our results can apply to or break in nonlinear networks and tasks, in the hope of inspiring future work.

5.1. Nonlinear Multimodal Networks

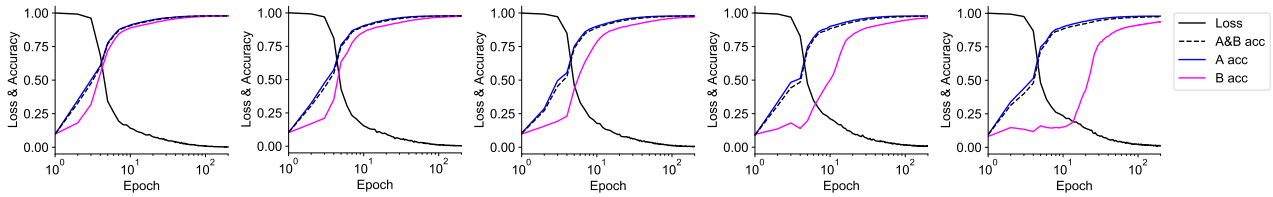
Linear Task. We find that our results, derived for linear networks, carry over to two-layer ReLU networks when the target task is linear. This aligns with the intuitions from a line of studies on the implicit bias of two-layer ReLU networks (Sarussi et al., 2021; Phuong & Lampert, 2021; Timor et al., 2023; Min et al., 2024). We simulate two-layer early and late fusion ReLU networks to learn a linear target map. As shown in Figure 11, the loss and weight trajectories are qualitatively the same as their linear counterparts in Figure 2, except that learning is about two times slower and the converged total weights are two times larger. We conduct simulations on the duration of the unimodal phase and the amount of mis-attribution in two-layer late fusion ReLU networks with the rest of the setting unchanged (target map is linear). The time ratio and mis-attribution in ReLU networks closely follow the theoretical predictions derived for linear networks as shown in Figures 3c and 3d (crosses).

Realistic Task. We validate our results in multimodal deep ReLU networks trained on a noisy MNIST (Lecun et al., 1998) task. In the noisy MNIST classification task shown in Figure 13, the multimodal network receives two MNIST written digit images as input, one of which may be corrupted by Gaussian noise. This is a common scenario in multimodal learning where the dominating modality varies per sample. We train multimodal ReLU networks with $L = 5$ and $L_f = 1, 2, \cdots, 5$ on this task. The multimodal ReLU networks learn the modality that is corrupted less often faster. We record the ratio between the time when the unimodal accuracy of the first learned modality reaches 50% and the time of the second modality. Table 1 records the mean and standard deviation of the time ratio across five random seeds. As shown in Figure 7 and Table 1, the deeper the layer at which fusion occurs, the slower the network learns from the second modality. This is consistent with the qualitative conclusion we draw from multimodal linear networks. Similar results are observed in multimodal convolutional networks, as shown in Figure 14 and Table 1.

Table 1. Time ratio in noisy MNIST digit classification.

Network	MLP	CNN
$L_f = 1$	1.30±0.24	1.04±0.03
$L_f = 2$	1.53±0.40	1.10±0.05
$L_f = 3$	1.90±0.39	1.08±0.03
$L_f = 4$	3.38±1.13	1.32±0.16
$L_f = 5$	6.38±1.12	2.53±0.45

Heterogeneous Task. We present a simple heterogeneous task that involves behaviors not observed in linear networks



(a) $L_f = 1$ (b) $L_f = 2$ (c) $L_f = 3$ (d) $L_f = 4$ (e) $L_f = 5$

Figure 7. Loss and accuracy trajectories of multimodal fully-connected ReLU networks. Five-layer ReLU newtorks with different fusion layer depths are trained on the noisy MNIST dataset. Solid black curves plot the training loss. Dotted black curves plot the multimodal classification accuracy, where both branches are presented with an uncorrupted testset image. Blue curves plot the unimodal classification accuracy of modality A, where branch A is presented with an uncorrupted testset image and branch B with a blank image. Similarly, for modality B (purple curves). The trajectories are averaged over five random seeds. Experimental details are provided in Appendix I.

and linear tasks. Consider learning $y = \boldsymbol{x}_{\mathrm{A}} + \mathrm{XOR}(\boldsymbol{x}_{\mathrm{B}})$, where $\boldsymbol{x}_{\mathrm{A}} \in \mathbb{R}, \boldsymbol{x}_{\mathrm{B}} \in \{[1,1],[1,-1],[-1,1],[-1,-1]\}$ and $\mathrm{XOR}(\boldsymbol{x}_{\mathrm{B}})$ refers to performing XOR to the two dimensions of $\boldsymbol{x}_{\mathrm{B}}$. We observe that two-layer late fusion ReLU networks always learn this task successfully, forming the four perpendicular XOR features as shown in Figures 12b, 12d and 12f. However, two-layer early fusion ReLU networks do not learn consistent XOR features and can even fail to learn this task as shown in Figures 12a, 12c and 12e. In the failed cases, the variance of $\boldsymbol{x}_{\mathrm{A}}$ is large so that the network can be stuck at a local minimum where the network only exploits the linear modality. For this heterogeneous task, late fusion networks are advantageous in terms of extracting heterogeneous features from each input modality. We provide videos of the learning dynamics on our website.

5.2. Practical Insights

We hypothesize that the practical choice of the fusion layer depth is a trade-off between alleviating unimodal bias and learning unimodal features. A shallower fusion layer helps alleviate unimodal bias because modalities can cooperate reciprocally to learn the synergistic computation. A deeper fusion layer helps unimodal feature learning because the network can operate more independently to learn the unique computation of extracting heterogeneous features from each modality. We hope our work contributes to a better understanding of this trade-off, ultimately leading to more systematic architectural choices and improved multimodal learning algorithms.

Acknowledgements

We thank Loek van Rossem and William Dorrell for feedback on a draft of this paper. We thank the following funding sources: Gatsby Charitable Foundation to YZ, PEL, and AS; Wellcome Trust (110114/Z/15/Z) to PEL; Sainsbury Wellcome Centre Core Grant from Wellcome (219627/Z/19/Z) to AS; a Sir Henry Dale Fellowship from the Wellcome Trust and Royal Society (216386/Z/19/Z) to AS.

Impact Statement

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

References

Advani, M. S., Saxe, A. M., and Sompolinsky, H. 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.

Agrawal, A., Batra, D., and Parikh, D. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1203. URL https://aclanthology.org/D16-1203.

Agrawal, A., Batra, D., Parikh, D., and Kembhavi, A. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In Dy, J. and Krause, A. (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.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkMQg3C5K7.

Atanasov, A., Bordelon, B., and Pehlevan, C. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607.

Barnum, G., Talukder, S. J., and Yue, Y. On the benefits of early fusion in multimodal representation learning. In *NeurIPS 2020 Workshop SVRHM*, 2020. URL https://openreview.net/forum?id=FMJ5e0IoFFk.

Braun, L., Dominé, C., Fitzgerald, J., and Saxe, A. Exact learning dynamics of deep linear networks with prior knowledge. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 6615–6629. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2b3bb2c95195130977a51b3bb251c40a-Paper-Conference.pdf.

Cadene, R., Dancette, C., Ben younes, H., Cord, M., and Parikh, D. Rubi: Reducing unimodal biases for visual question answering. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/51d92be1c60d1db1d2e5e7a07da55b26-Paper.pdf.

Du, C., Teng, J., Li, T., Liu, Y., Yuan, T., Wang, Y., Yuan, Y., and Zhao, H. On uni-modal feature learning in supervised multi-modal learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8632–8656. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/du23e.html.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/fe131d7f5a6b38b23cc967316c13dae2-Paper.pdf.

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

Gadzicki, K., Khamsehashari, R., and Zetzsche, C. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–6, 2020. doi: 10.23919/FUSION45008.2020.9190246.

Gat, I., Schwartz, I., Schwing, A., and Hazan, T. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3197–3208. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/20d749bc05f47d2bd3026ce457dcfd8e-Paper.pdf.

Gat, I., Schwartz, I., and Schwing, A. Perceptual score: What data modalities does your model perceive? In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21630–21643. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b51a15f382ac914391a58850ab343b00-Paper.pdf.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (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.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Hessel, J. and Lee, L. Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pp. 861–877, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.62. URL https://aclanthology.org/2020.emnlp-main.62.

Huang, Y., Du, C., Xue, Z., Chen, X., Zhao, H., and Huang, L. What makes multi-modal learning better than single (provably). In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10944–10956. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/5aa3405a3f865c10f420a4a7b55cbff3-Paper.pdf.

Huang, Y., Lin, J., Zhou, C., Yang, H., and Huang, L. Modality competition: What makes joint training of multimodal network fail in deep learning? (Provably). In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9226–9259. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/huang22e.html.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Huh, D. Curvature-corrected learning dynamics in deep neural networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4552–4560. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/huh20a.html.

Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJflg30qKX.

Kleinman, M., Achille, A., and Soatto, S. Critical learning periods for multisensory integration in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24296–24305, June 2023.

Krogh, A. and Hertz, J. A. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General*, 25(5):1135, March 1992. doi: 10.1088/0305-4470/25/5/020. URL https://dx.doi.org/10.1088/0305-4470/25/5/020.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryfMLoCqtQ.

Lao, M., Guo, Y., Liu, Y., Chen, W., Pu, N., and Lew, M. S. From superficial to deep: Language bias driven curriculum learning for visual question answering. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pp. 3370–3379, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475492. URL https://doi.org/10.1145/3474085.3475492.

Le Cun, Y., Kanter, I., and Solla, S. A. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66:2396–2399, May 1991. doi: 10.1103/PhysRevLett.66.2396. URL https://link.aps.org/doi/10.1103/PhysRevLett.66.2396.

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

Liang, P. P., Zadeh, A., and Morency, L.-P. Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. *ACM Comput. Surv.*, apr 2024. ISSN 0360-0300. doi: 10.1145/3656580. URL https://doi.org/10.1145/3656580.

Makino, T., Wang, Y., Geras, K. J., and Cho, K. Detecting incidental correlation in multimodal learning via latent variable modeling. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=QoRo9QmOAr.

Min, H., Mallada, E., and Vidal, R. Early neuron alignment in two-layer relu networks with small initialization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QibPzdVrRu.

Peng, X., Wei, Y., Deng, A., Wang, D., and Hu, D. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8238–8247, June 2022.

Pesme, S. and Flammarion, N. Saddle-to-saddle dynamics in diagonal linear networks. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7475–7505. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/17a9ab4190289f0e1504bbb98d1d111a-Paper-Conference.pdf.

Phuong, M. and Lampert, C. H. The inductive bias of relu networks on orthogonally separable data. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=krz7T0xU9Z_.

Ramachandram, D. and Taylor, G. W. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017. doi: 10.1109/MSP.2017.2738401.

Ramakrishnan, S., Agrawal, A., and Lee, S. Overcoming language priors in visual question answering with adversarial regularization. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.

Sarussi, R., Brutzkus, A., and Globerson, A. Towards understanding learning in neural networks with linear teachers. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9313–9322. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/sarussi21a.html.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6120.

Saxe, A. M., McClelland, J. L., and Ganguli, S. 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.

Shi, J., Shea-Brown, E., and Buice, M. Learning dynamics of deep linear networks with multiple pathways. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34064–34076. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/dc3ca8bcd613e43ce540352b58d55d6d-Paper-Conference.pdf.

Tarmoun, S., Franca, G., Haeffele, B. D., and Vidal, R. Understanding the dynamics of gradient flow in overparameterized linear models. In Meila, M. and Zhang, T. (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.

Timor, N., Vardi, G., and Shamir, O. Implicit regularization towards rank minimization in relu networks. In Agrawal, S. and Orabona, F. (eds.), *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pp. 1429–1459. PMLR, 20 Feb–23 Feb 2023.

Wang, W., Tran, D., and Feiszli, M. What makes training multi-modal classification networks hard? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12692–12702, 2020. doi: 10.1109/CVPR42600.2020.01271.

Wu, N., Jastrzebski, S., Cho, K., and Geras, K. J. Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24043–24055. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wu22d.html.

Xie, B., Liang, Y., and Song, L. Diverse Neural Network Learns True Target Functions. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1216–1224. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/xie17a.html.

Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., and Parikh, D. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Appendix

A. Additional Related Work

A.1. Technical Challenges in Multimodal Deep Linear Networks

There are two main challenges of studying multimodal deep linear networks, compared with standard deep linear networks.

The first challenge is that multimodal networks are not fully-connected. The gradient descent dynamics of multimodal linear networks derived in Appendix B differ from those in fully-connected linear networks. One complication is that the standard balancing property (Du et al., 2018) does not hold across all layers. In fully-connected linear networks, all layers maintain balance and have equal norm throughout learning (Du et al., 2018). However, in multimodal linear networks, post-fusion layers are balanced and pre-fusion layers of the two modalities are respectively balanced, as discussed in Appendix E.1. Thus, the dynamics of multimodal linear networks are summarized by three variables (post-fusion layers, pre-fusion layers of modality A, and pre-fusion layers of modality B), in contrast to one variable in fully-connected linear networks.

The second challenge is that multimodal datasets generally do not have whitened input. A considerable portion of theoretical work on linear network learning dynamics (Fukumizu, 1998; Saxe et al., 2014; 2019; Lampinen & Ganguli, 2019; Arora et al., 2019; Huh, 2020; Tarmoun et al., 2021; Atanasov et al., 2022; Braun et al., 2022) assumes whitened input covariance matrix, i.e. identity matrix. With the whitened input assumption, the learning dynamics of linear networks with small or balanced initialization can be written as a Riccati equation, which has known closed-form solutions (Fukumizu, 1998; Saxe et al., 2014; Tarmoun et al., 2021; Shi et al., 2022; Braun et al., 2022). However, we did not use this assumption because it is unrealistic to assume that multimodal data streams are whitened. For the unwhitened input case, existing literature does not provide closed-form solutions and Tarmoun et al. (2021) stated that they believe the dynamics cannot be solved exactly. Given the unavailability of closed-form solutions, we conduct a fixed point analysis and a calculation of the time ratio to gain insights relevant to unimodal bias. Our fixed point analysis is a standard tool for dynamical systems but has not been done for multimodal deep linear networks before. Our calculation of the time presents novel tools to understand the timing of phase transitions in linear networks.

A.2. Alleviating Unimodal Bias in Practice

Unimodal bias is a common issue in multimodal learning. Practitioners have proposed many approaches to alleviate the bias.

Some approaches construct more balanced datasets to alleviate the language bias in Visual Question Answering (VQA). Zhang et al. (2016) added training samples with the opposite answer for binary VQA datasets. Goyal et al. (2017) built VQA v2 dataset by adding images with different answers into the popular VQA v1 dataset (Antol et al., 2015). Agrawal et al. (2018) presented new splits of the VQA v1 and VQA v2 datasets in which the train and test sets have different prior distributions of answers. The new splits can be used to diagnose the unimodal bias in VQA models.

Some approaches alter the training process of multimodal models to alleviate the unimodal bias. Ramakrishnan et al. (2018) employed adversarial training for VQA models. They posed training as an adversarial game between the VQA model and a question-only model, discouraging the VQA model from capturing language biases. Lao et al. (2021) employed curriculum learning for VQA models. Their VQA model learns from the language modality first and then progressively learns multimodal reasoning from less-biased samples. RUBi (Cadene et al., 2019) leverages a question-only model to dynamically adjust the loss in the training of VQA models. Gat et al. (2020) proposed a new regularization term based on the functional entropy, which encourages the multimodal model to balance the contribution of each modality in training. Wang et al. (2020); Peng et al. (2022) inspect the contribution of each modality in the multimodal model and dynamically re-weight the loss (Wang et al., 2020) or the gradient (Peng et al., 2022) to balance the modalities during training. Wu et al. (2022) inspect the speed at which the model learns from each modality and accelerate learning from underutilized modalities. Du et al. (2023) introduced knowledge distillation to multimodel learning. They distill pretrained unimodal features into corresponding parts of the late fusion multimodal model to facilitate unimodal feature learning.

B. Gradient Descent in Deep Multimodal Linear Networks

We derive the gradient descent dynamics in multimodal deep linear networks with learning rate η. In pre-fusion layers $1 \leq l \leq L_f$, the gradient update is

$$\begin{aligned}
\Delta \boldsymbol{W}_{\mathrm{A}}^{l} &= -\eta \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \boldsymbol{W}_{\mathrm{A}}^{l}} \\
&= \eta \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^{j} \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} (y - \hat{y}) \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{A}}^{i} \boldsymbol{x}_{\mathrm{A}} \right)^{\top} \\
&= \eta \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^{j} \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} \left(y - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{A}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{B}} \right) \boldsymbol{x}_{\mathrm{A}}^{\top} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} \\
&= \eta \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^{j} \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} \left(\boldsymbol{\Sigma}_{y \boldsymbol{x}_{\mathrm{A}}} - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{A}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{BA}} \right) \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} \qquad (14a) \\
\Delta \boldsymbol{W}_{\mathrm{B}}^{l} &= \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^{j} \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^{i} \right)^{\top} \left(\boldsymbol{\Sigma}_{y \boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{BA}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{B}} \right) \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{B}}^{i} \right)^{\top} \qquad (14b)
\end{aligned}$$

In post-fusion layers $L_f + 1 \leq l \leq L$,

$$\begin{aligned}
\Delta \boldsymbol{W}^{l} &= -\eta \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \boldsymbol{W}^{l}} \\
&= \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} (y - \hat{y}) \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \boldsymbol{x}_{\mathrm{A}} \right)^{\top} \\
&+ \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} (y - \hat{y}) \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^{i} \boldsymbol{x}_{\mathrm{B}} \right)^{\top} \\
&= \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} \left(y - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{A}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{B}} \right) \boldsymbol{x}_{\mathrm{A}}^{\top} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \right)^{\top} \\
&+ \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} \left(y - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{A}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{x}_{\mathrm{B}} \right) \boldsymbol{x}_{\mathrm{B}}^{\top} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^{i} \right)^{\top} \\
&= \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} \left(\boldsymbol{\Sigma}_{y \boldsymbol{x}_{\mathrm{A}}} - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{A}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{BA}} \right) \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^{i} \boldsymbol{x}_{\mathrm{A}} \right)^{\top} \\
&+ \eta \left(\prod_{j=l+1}^{L} \boldsymbol{W}^{j} \right)^{\top} \left(\boldsymbol{\Sigma}_{y \boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{AB}} - \boldsymbol{W}_{\mathrm{B}}^{\mathrm{tot}} \boldsymbol{\Sigma}_{\mathrm{B}} \right) \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^{j} \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^{i} \boldsymbol{x}_{\mathrm{B}} \right)^{\top} \qquad (15)
\end{aligned}$$

In the limit of small learning rate, the difference equations Equations (14) and (15) are well approximated by the differential equations Equations (3) and (4) in the main text.

C. Two-Layer Early Fusion Linear Network

A two-layer early fusion linear network is described as $\hat{y} = \boldsymbol{W}^2\boldsymbol{W}^1\boldsymbol{x}$. The gradient descent dynamics are

$$\tau\dot{\boldsymbol{W}}^1 = {\boldsymbol{W}^2}^\top(\boldsymbol{\Sigma}_{y\boldsymbol{x}} - \boldsymbol{W}^2\boldsymbol{W}^1\boldsymbol{\Sigma}), \quad \tau\dot{\boldsymbol{W}}^2 = (\boldsymbol{\Sigma}_{y\boldsymbol{x}} - \boldsymbol{W}^2\boldsymbol{W}^1\boldsymbol{\Sigma}){\boldsymbol{W}^1}^\top. \tag{16}$$

By setting to the gradients to zero, we find that there are two manifolds of fixed points:

$$\boldsymbol{W}^2 = \boldsymbol{0},\ \boldsymbol{W}^1 = \boldsymbol{0} \quad \Rightarrow \quad \mathcal{M}_0 = \{\boldsymbol{W}|\boldsymbol{W} = \boldsymbol{0}\}, \tag{17a}$$

$$\boldsymbol{\Sigma}_{y\boldsymbol{x}} - \boldsymbol{W}^2\boldsymbol{W}^1\boldsymbol{\Sigma} = \boldsymbol{0} \quad \Rightarrow \quad \mathcal{M}_* = \{\boldsymbol{W}|\boldsymbol{W}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}\}. \tag{17b}$$

We plot $\mathcal{M}_0, \mathcal{M}_*$ in Figure 2c for a scalar case introduced in Figure 2b.

D. Two-Layer Late Fusion Linear Network

A two-layer late fusion linear network is described as $\hat{y} = \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{x}_{\text{A}} + \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{x}_{\text{B}}$. The gradient descent dynamics are

$$\tau\dot{\boldsymbol{W}}_{\text{A}}^1 = {\boldsymbol{W}_{\text{A}}^2}^\top(\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{A}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{BA}}), \tag{18a}$$

$$\tau\dot{\boldsymbol{W}}_{\text{A}}^2 = (\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{A}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{BA}}){\boldsymbol{W}_{\text{A}}^1}^\top, \tag{18b}$$

$$\tau\dot{\boldsymbol{W}}_{\text{B}}^1 = {\boldsymbol{W}_{\text{B}}^2}^\top(\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{AB}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{B}}), \tag{18c}$$

$$\tau\dot{\boldsymbol{W}}_{\text{B}}^2 = (\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{AB}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{B}}){\boldsymbol{W}_{\text{B}}^1}^\top. \tag{18d}$$

D.1. Fixed Points and Saddles

By setting the gradients in Equation (18) to zero, we find that the two manifolds of fixed points in Equation (6) exist in two-layer late fusion linear networks as well.

$$\begin{cases} \boldsymbol{W}_{\text{A}}^2 = \boldsymbol{0},\ \boldsymbol{W}_{\text{A}}^1 = \boldsymbol{0} \\ \boldsymbol{W}_{\text{B}}^2 = \boldsymbol{0},\ \boldsymbol{W}_{\text{B}}^1 = \boldsymbol{0} \end{cases} \quad \Rightarrow \quad \mathcal{M}_0 = \{\boldsymbol{W}|\boldsymbol{W} = \boldsymbol{0}\}, \tag{19a}$$

$$\begin{cases} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{A}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{BA}} = \boldsymbol{0} \\ \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{AB}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{B}} = \boldsymbol{0} \end{cases} \quad \Rightarrow \quad \mathcal{M}_* = \{\boldsymbol{W}|\boldsymbol{W}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}\}. \tag{19b}$$

In addition, there are two manifolds of saddles:

$$\begin{cases} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}} - \boldsymbol{W}_{\text{A}}^2\boldsymbol{W}_{\text{A}}^1\boldsymbol{\Sigma}_{\text{A}} = \boldsymbol{0} \\ \boldsymbol{W}_{\text{B}}^2 = \boldsymbol{0},\ \boldsymbol{W}_{\text{B}}^1 = \boldsymbol{0} \end{cases} \quad \Rightarrow \quad \mathcal{M}_{\text{A}} = \{\boldsymbol{W}|\boldsymbol{W}_{\text{A}}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}}{\boldsymbol{\Sigma}_{\text{A}}}^{-1}, \boldsymbol{W}_{\text{B}}^{\text{tot}} = \boldsymbol{0}\}, \tag{20a}$$

$$\begin{cases} \boldsymbol{W}_{\text{A}}^2 = \boldsymbol{0},\ \boldsymbol{W}_{\text{A}}^1 = \boldsymbol{0} \\ \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}} - \boldsymbol{W}_{\text{B}}^2\boldsymbol{W}_{\text{B}}^1\boldsymbol{\Sigma}_{\text{B}} = \boldsymbol{0} \end{cases} \quad \Rightarrow \quad \mathcal{M}_{\text{B}} = \{\boldsymbol{W}|\boldsymbol{W}_{\text{A}}^{\text{tot}} = \boldsymbol{0}, \boldsymbol{W}_{\text{B}}^{\text{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}}{\boldsymbol{\Sigma}_{\text{B}}}^{-1}\}. \tag{20b}$$

We plot the four manifolds $\mathcal{M}_0, \mathcal{M}_*, \mathcal{M}_{\text{A}}, \mathcal{M}_{\text{B}}$ in Figure 2f for a scalar case introduced in Figure 2e.

D.2. A Solvable Simple Case

If the two modalities are not correlated ($\boldsymbol{\Sigma}_{\text{AB}} = \boldsymbol{0}$) and have white covariance ($\boldsymbol{\Sigma}_{\text{A}} = \sigma_{\text{A}}^2\boldsymbol{I}, \boldsymbol{\Sigma}_{\text{B}} = \sigma_{\text{B}}^2\boldsymbol{I}$) the dynamics is equivalent to two separately trained unimodal two-layer linear networks with whitened input, whose solution has been derived in Saxe et al. (2014). The time course solutions of total weights are

$$\boldsymbol{W}_{\text{A}}^{\text{tot}}(t) = u_{\text{A}}(t)\sigma_{\text{A}}^{-2}\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}},\ u_{\text{A}}(t) = \left[\left(\frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}}\|}{\sigma_{\text{A}}^2 u_{\text{A}}(0)} - 1\right)e^{-2\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}}\|\frac{t}{\tau}} + 1\right]^{-1}, \tag{21a}$$

$$\boldsymbol{W}_{\text{B}}^{\text{tot}}(t) = u_{\text{B}}(t)\sigma_{\text{B}}^{-2}\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}},\ u_{\text{B}}(t) = \left[\left(\frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}}\|}{\sigma_{\text{B}}^2 u_{\text{B}}(0)} - 1\right)e^{-2\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}}\|\frac{t}{\tau}} + 1\right]^{-1}. \tag{21b}$$

The total weights for both modalities only evolve in scale along the pseudo-inverse solution direction. The scale variables $u_{\text{A}}(t), u_{\text{B}}(t)$ both go through the sigmoidal growth while modality A grows approximately $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{A}}}\|/\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\text{B}}}\|$ times faster.

D.3. General Cases

In the general case of having arbitrary correlation matrices, the analytical solution in Equation (21) does not hold since the total weights not only grow in one fixed direction but also rotate. We then focus on the early phase dynamics to compute the duration of the unimodal phase.

D.3.1. Which Modality is Learned First and When?

We assume that the modality learned first is modality A and specify the time t_{A} and the condition for prioritizing modality A in the following. During time 0 to t_{A}, the total weights of both modalities have not moved much away from their small initialization near zero. Hence, the dynamics from time 0 to t_{A} in leading order approximation of the initialization are

$$\tau \dot{\boldsymbol{W}}_{\mathrm{A}}^{1} = \boldsymbol{W}_{\mathrm{A}}^{2\top} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}},\ \tau \dot{\boldsymbol{W}}_{\mathrm{A}}^{2} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{W}_{\mathrm{A}}^{1\top}; \tag{22a}$$

$$\tau \dot{\boldsymbol{W}}_{\mathrm{B}}^{1} = \boldsymbol{W}_{\mathrm{B}}^{2\top} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}},\ \tau \dot{\boldsymbol{W}}_{\mathrm{B}}^{2} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{W}_{\mathrm{B}}^{1\top}. \tag{22b}$$

The first-layer weights $\boldsymbol{W}_{\mathrm{A}}^{1}, \boldsymbol{W}_{\mathrm{B}}^{1}$ align with $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}, \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}$ respectively in the early phase when their scale has not grown appreciably (Atanasov et al., 2022). We have the balancing property (Du et al., 2018; Ji & Telgarsky, 2019) between the two layers of modality A

$$\boldsymbol{W}_{\mathrm{A}}^{1} \boldsymbol{W}_{\mathrm{A}}^{1\top} = \boldsymbol{W}_{\mathrm{A}}^{2\top} \boldsymbol{W}_{\mathrm{A}}^{2} \quad \Rightarrow \quad \|\boldsymbol{W}_{\mathrm{A}}^{1}\|_{\mathrm{F}} = \|\boldsymbol{W}_{\mathrm{A}}^{2}\| \stackrel{\text{def}}{=} u_{\mathrm{A}}. \tag{23}$$

We conduct the following change of variable

$$\boldsymbol{W}_{\mathrm{A}}^{1} = u_{\mathrm{A}}(t) \boldsymbol{r}_{\mathrm{A}}^{1} \frac{\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|},\ \boldsymbol{W}_{\mathrm{A}}^{2} = u_{\mathrm{A}}(t) \boldsymbol{r}_{\mathrm{A}}^{1\top}, \tag{24}$$

where $\boldsymbol{r}_{\mathrm{A}}^{1}$ is a fixed unit norm column vector representing the freedom in the hidden layer and u_{A} is a scalar representing the norm of the two balancing layers (Advani et al., 2020). Substituting the variables, we can re-write the dynamics of $\boldsymbol{W}_{\mathrm{A}}^{2}$ and obtain an ordinary differential equation about u_{A}:

$$\tau \dot{u}_{\mathrm{A}} \boldsymbol{r}_{\mathrm{A}}^{1\top} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \frac{\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}^{\top}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|} \boldsymbol{r}_{\mathrm{A}}^{1\top} u_{\mathrm{A}} \quad \Rightarrow \quad \tau \dot{u}_{\mathrm{A}} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| u_{\mathrm{A}}. \tag{25}$$

Separating variables and integrating both sides, we get

$$t = \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} (\ln u_{\mathrm{A}} - \ln u_0), \tag{26}$$

where u_0 denotes $u_0 = u_{\mathrm{A}}(0)$. Because the initialization u_0 is very small compared to $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|$, the time that u_{A} grows to be comparable with $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|$ is

$$t_{\mathrm{A}} \approx \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} \ln \frac{1}{u_0}, \tag{27}$$

From Equation (27), we can infer that the condition for modality A to be learned first is that $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| > \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|$.

$$\tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} \ln \frac{1}{u_0} < \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|^{-1} \ln \frac{1}{u_0} \quad \Leftrightarrow \quad \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| > \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|. \tag{28}$$

D.3.2. When is the Second Modality Learned?

We next calculate the time t_{B} when modality B is learned. During time 0 to t_{A}, the weight dynamics of modality B takes the same form as modality A as described in Equation (22b). Because the layers in the modality B branch are balanced and $\boldsymbol{W}_{\mathrm{B}}^{1}$ aligns with the rank-one direction $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}$, we again change variables and re-write the dynamics of $\boldsymbol{W}_{\mathrm{B}}^{2}$, obtaining an ordinary differential equation about the norm of the two balanced layers $u_{\mathrm{B}} = \|\boldsymbol{W}_{\mathrm{B}}^{1}\|_{\mathrm{F}} = \|\boldsymbol{W}_{\mathrm{B}}^{2}\|$:

$$\tau \dot{u}_{\mathrm{B}} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\| u_{\mathrm{B}} \quad \Rightarrow \quad t = \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|^{-1} (\ln u_{\mathrm{B}} - \ln u_0),\ t \in [0, t_{\mathrm{A}}). \tag{29}$$

We assume $u_{\mathrm{B}}(0) = u_0$, since the initialization is small and the number of hidden neurons in the modality B branch is of the same order as modality A. We plug t_{A}, obtained in Equation (27), into Equation (29) and get

$$\ln u_{\mathrm{B}}(t_{\mathrm{A}}) = \left(1 - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1}\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|\right) \ln u_0. \tag{30}$$

We then look into the dynamics during the unimodal phase from t_{A} to t_{B}. During time t_{A} to t_{B}, the weights of modality B are still small and negligible in leading ordering approximation. Meanwhile, the weights of modality A have grown to be $\boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}$, which influences the dynamics of modality B when the cross-covariance $\boldsymbol{\Sigma}_{\mathrm{AB}} \neq \boldsymbol{0}$. Taking this into consideration, the dynamics of modality B during t_{A} to t_{B} is

$$\tau \dot{\boldsymbol{W}}_{\mathrm{B}}^1 = {\boldsymbol{W}_{\mathrm{B}}^2}^\top \widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}},\ \tau \dot{\boldsymbol{W}}_{\mathrm{B}}^2 = \widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}} {\boldsymbol{W}_{\mathrm{B}}^1}^\top,\ t \in (t_{\mathrm{A}}, t_{\mathrm{B}}), \tag{31}$$

where $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}$. The first-layer weights $\boldsymbol{W}_{\mathrm{B}}^1$ rapidly rotate from $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}$ to $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}$ at time t_{A} and continue to evolve along $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}$ during t_{A} to t_{B}. Through the same manner of changing variables, we obtain the ordinary differential equation about u_{B} during t_{A} to t_{B}:

$$\tau \dot{u}_{\mathrm{B}} = \left\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\right\| u_{\mathrm{B}} \quad \Rightarrow \quad t - t_{\mathrm{A}} = \tau \left\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\right\|^{-1} (\ln u_{\mathrm{B}} - \ln u_{\mathrm{B}}(t_{\mathrm{A}})),\ t \in (t_{\mathrm{A}}, t_{\mathrm{B}}). \tag{32}$$

Plugging in $u_{\mathrm{B}}(t_{\mathrm{A}})$ obtained in Equation (30), we get the time when u_{B} grows to be comparable with $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|$:

$$t_{\mathrm{B}} \approx t_{\mathrm{A}} - \tau \left\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\right\|^{-1} \ln u_{\mathrm{B}}(t_{\mathrm{A}}) \approx t_{\mathrm{A}} + \tau \frac{1 - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1}\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\right\|} \ln \frac{1}{u_0}. \tag{33}$$

Dividing Equation (33) by Equation (27), we obtain the time ratio Equation (9) in the main text:

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} \approx 1 + \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|}.$$

We validate Equation (9) with numerical simulations in Figure 3c. Note that if $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ have collinearity, the denominator in Equation (9) is zero and the time ratio is ∞. As shown in Figure 8, the late fusion network learns to fit the output only with modality A and modality B will never be learned.

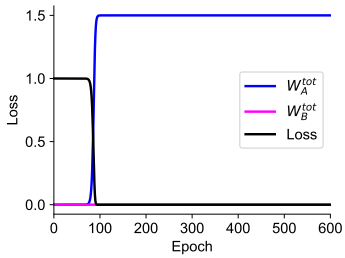
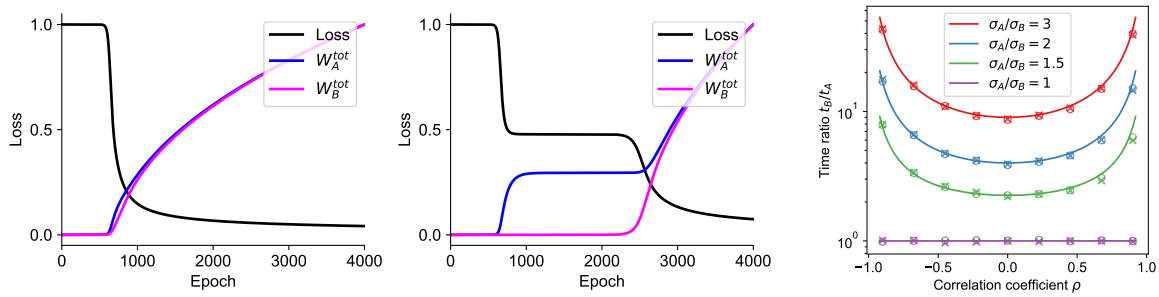


Figure 8. Loss and weight trajectories in a two-layer late fusion network when modalities have collinearity. Inputs $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ are scalars with covariance matrix $\boldsymbol{\Sigma} = [4, 2; 2, 1]$. The target output is generated as $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$.

D.4. Logistic Loss

Our qualitative and quantitative results, derived with mean square error loss, carry over to multimodal deep linear network with logistic loss. As shown in Figure 9, the early fusion linear network trained with logistic loss learns both modality almost simultaneously while the late fusion network learns two modalities at two separate times, which is similar to Figures 2b and 2e. Note that weights diverge due to the nature of the logistic loss, which cannot attain zero with any finite weights and keeps decreasing for larger $\hat{y}$ values with the correct sign.

The time ratio in late fusion linear networks trained with logistic loss is the same as those trained with mean square error loss as demonstrated in Figure 9c. This is because the first order approximation of the gradient descent dynamics in the early phase of training is the same for the two loss functions.



(a) Early fusion (b) Late fusion (c) Time ratio in late fusion

Figure 9. Two-layer early and late fusion linear networks trained with logistic loss. (a,b) Loss and weights trajectories. (c) Time ratio as in Equation (9).

In the early phase of training, the network output $\hat{y}$ is close to 0 since we assume small initialization. Thus the Taylor expansion of the loss function around $\hat{y} = 0$ is a good approximation in the early phase

$$\ell_{\text{MSE}}(\hat{y}) \equiv \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}y^2 - y\hat{y} + O(\hat{y}^2), \tag{34}$$

$$\ell_{\text{LG}}(\hat{y}) \equiv \ln\left(1 + \exp(-y^\mu \hat{y}^\mu)\right) = \ell_{\text{LG}}(0) + \ell'_{\text{LG}}(0)\hat{y} + O(\hat{y}^2) = \ln 2 - \frac{1}{2}y\hat{y} + O(\hat{y}^2). \tag{35}$$

We can use the first-order expansion to write the gradient descent dynamics in the early phase

$$\tau \dot{\boldsymbol{W}} = -\frac{d\ell_{\text{MSE}}}{d\hat{y}}\frac{d\hat{y}}{d\boldsymbol{W}} \approx y\frac{d\hat{y}}{d\boldsymbol{W}}, \tag{36}$$

$$\tau \dot{\boldsymbol{W}} = -\frac{d\ell_{\text{LG}}}{d\hat{y}}\frac{d\hat{y}}{d\boldsymbol{W}} \approx \frac{1}{2}y\frac{d\hat{y}}{d\boldsymbol{W}}, \tag{37}$$

which are the same for square and logistic loss except a constant factor. Since the calculation of the time ratio mainly involves the early phase dynamics, the time ratio in late fusion linear networks remains the same whether trained with mean square error or logistic loss.

E. Deep Intermediate & Late Fusion Linear Networks

E.1. Balancing Properties

The full dynamics of deep intermediate and late fusion linear networks are given in Equations (3) and (4). Between pre-fusion layers and between post-fusion layers, the standard balancing property (Du et al., 2018; Ji & Telgarsky, 2019) holds true:

$$\boldsymbol{W}_{\text{A}}^1 \boldsymbol{W}_{\text{A}}^{1\top} = \boldsymbol{W}_{\text{A}}^2 \boldsymbol{W}_{\text{A}}^{2\top} = \cdots = \boldsymbol{W}_{\text{A}}^{L_f} \boldsymbol{W}_{\text{A}}^{L_f\top}, \tag{38a}$$

$$\boldsymbol{W}_{\text{B}}^1 \boldsymbol{W}_{\text{B}}^{1\top} = \boldsymbol{W}_{\text{B}}^2 \boldsymbol{W}_{\text{B}}^{2\top} = \cdots = \boldsymbol{W}_{\text{B}}^{L_f} \boldsymbol{W}_{\text{B}}^{L_f\top}, \tag{38b}$$

$$\boldsymbol{W}^{L_f+1} \boldsymbol{W}^{L_f+1\top} = \boldsymbol{W}^{L_f+2} \boldsymbol{W}^{L_f+2\top} = \cdots = \boldsymbol{W}^{L\top} \boldsymbol{W}^{L}. \tag{38c}$$

Between a pre-fusion layer and a post-fusion layer, the balancing condition takes a slightly different form:

$$\boldsymbol{W}_{\text{A}}^{L_f} \boldsymbol{W}_{\text{A}}^{L_f\top} + \boldsymbol{W}_{\text{B}}^{L_f} \boldsymbol{W}_{\text{B}}^{L_f\top} = \boldsymbol{W}^{L_f+1} \boldsymbol{W}^{L_f+1\top}. \tag{39}$$

Based on the standard balancing properties in Equation (38), we have equal norm between pre-fusion layers and between post-fusion layers.

$$\|\boldsymbol{W}_{\text{A}}^1\|_{\text{F}} = \|\boldsymbol{W}_{\text{A}}^2\|_{\text{F}} = \cdots = \|\boldsymbol{W}_{\text{A}}^{L_f}\|_{\text{F}} \stackrel{\text{def}}{=} u_{\text{A}}, \tag{40a}$$

$$\|\boldsymbol{W}_{\text{B}}^1\|_{\text{F}} = \|\boldsymbol{W}_{\text{B}}^2\|_{\text{F}} = \cdots = \|\boldsymbol{W}_{\text{B}}^{L_f}\|_{\text{F}} \stackrel{\text{def}}{=} u_{\text{B}}, \tag{40b}$$

$$\|\boldsymbol{W}^{L_f+1}\|_{\text{F}} = \|\boldsymbol{W}^{L_f+2}\|_{\text{F}} = \cdots = \|\boldsymbol{W}^{L}\| \stackrel{\text{def}}{=} u. \tag{40c}$$

Based on Equation (39), we have the balancing property between the norm of a pre-fusion layer and a post-fusion layer.

$$u_{\mathrm{A}}^2 + u_{\mathrm{B}}^2 = u^2. \tag{41}$$

In addition to the balancing norm, we infer from the balancing properties that post-fusion layers have rank-one structures as in standard linear networks (Ji & Telgarsky, 2019; Atanasov et al., 2022). The pre-fusion layers are not necessarily rank-one due to the different balancing property in Equation (39). However, guided by empirical observations, we make the ansatz that the weights in pre-fusion layers are also rank-one, which will enable us to conduct the change of variables similar to what we have done in Appendix D.3.

E.2. Time Ratio for $L_f \neq 2$

E.2.1. Which Modality is Learned First and When?

We adopt the convention that the modality learned first is modality A. During time 0 to t_{A}, the dynamics in leading order approximation of the initialization are

$$\tau \dot{\boldsymbol{W}}_{\mathrm{A}}^l = \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^j \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^i \right)^\top \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{A}}^i \right)^\top, \ 1 \leq l \leq L_f, \tag{42a}$$

$$\tau \dot{\boldsymbol{W}}_{\mathrm{B}}^l = \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^j \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^i \right)^\top \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{B}}^i \right)^\top, \ 1 \leq l \leq L_f, \tag{42b}$$

$$\begin{aligned} \tau \dot{\boldsymbol{W}}^l = &\left(\prod_{j=l+1}^{L} \boldsymbol{W}^j \right)^\top \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^j \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{A}}^i \right)^\top \\ &+ \left(\prod_{j=l+1}^{L} \boldsymbol{W}^j \right)^\top \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \left(\prod_{j=L_f+1}^{l-1} \boldsymbol{W}^j \prod_{i=1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^i \right)^\top, \ L_f + 1 \leq l \leq L. \end{aligned} \tag{42c}$$

We make the following change of variables

$$\boldsymbol{W}_{\mathrm{A}}^1 = u_{\mathrm{A}}(t) \boldsymbol{r}_{\mathrm{A}}^1 \frac{\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|}, \ \boldsymbol{W}_{\mathrm{B}}^1 = u_{\mathrm{B}}(t) \boldsymbol{r}_{\mathrm{B}}^1 \frac{\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}; \tag{43a}$$

$$\boldsymbol{W}_{\mathrm{A}}^i = u_{\mathrm{A}}(t) \boldsymbol{r}_{\mathrm{A}}^i \boldsymbol{r}_{\mathrm{A}}^{i-1\top}, \ \boldsymbol{W}_{\mathrm{B}}^i = u_{\mathrm{B}}(t) \boldsymbol{r}_{\mathrm{B}}^i \boldsymbol{r}_{\mathrm{B}}^{i-1\top}, \ 2 \leq i \leq L_f - 1; \tag{43b}$$

$$\boldsymbol{W}_{\mathrm{A}}^{L_f} = u_{\mathrm{A}}(t) \boldsymbol{r}^{L_f} \boldsymbol{r}_{\mathrm{A}}^{L_f-1\top}, \ \boldsymbol{W}_{\mathrm{B}}^{L_f} = u_{\mathrm{B}}(t) \boldsymbol{r}^{L_f} \boldsymbol{r}_{\mathrm{B}}^{L_f-1\top}; \tag{43c}$$

$$\boldsymbol{W}^j = u(t) \boldsymbol{r}^j \boldsymbol{r}^{j-1\top}, \ L_f + 1 \leq j \leq L - 1; \tag{43d}$$

$$\boldsymbol{W}^L = u(t) \boldsymbol{r}^{L\top}. \tag{43e}$$

where all $\boldsymbol{r}$ are fixed unit norm column vectors representing the freedom in hidden layers. By substituing Equation (43) into Equation (42), we reduce the dynamics during time 0 to t_{A} to a three-dimensional dynamical system about $u_{\mathrm{A}}, u_{\mathrm{B}}$ and u:

$$\tau \dot{u}_{\mathrm{A}} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| u_{\mathrm{A}}^{L_f-1} u^{L-L_f}, \tag{44a}$$

$$\tau \dot{u}_{\mathrm{B}} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\| u_{\mathrm{B}}^{L_f-1} u^{L-L_f}, \tag{44b}$$

$$\tau \dot{u} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| u_{\mathrm{A}}^{L_f} u^{L-L_f-1} + \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\| u_{\mathrm{B}}^{L_f} u^{L-L_f-1}. \tag{44c}$$

We divide Equation (44b) by Equation (44a) and reveal an equality between the two branches of a pre-fusion layer:

$$\frac{du_{\mathrm{B}}}{du_{\mathrm{A}}} = \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\| u_{\mathrm{B}}^{L_f-1}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| u_{\mathrm{A}}^{L_f-1}} \quad \Rightarrow \quad \frac{u_{\mathrm{B}}^{2-L_f} - u_0^{2-L_f}}{u_{\mathrm{A}}^{2-L_f} - u_0^{2-L_f}} = \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|} \stackrel{\text{def}}{=} k, \ L_f \neq 2, \tag{45}$$

We assume $L_f \neq 2$ in this section and handle the case of $L_f = 2$ in Appendix E.3. By utilizing two equality properties in Equation (45) and Equation (41), we reduce Equation (44a) to an ordinary differential equation about u_{A}

$$\tau \dot{u}_{\mathrm{A}} = \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| u_{\mathrm{A}}^{L_f-1} \left[u_{\mathrm{A}}^2 + \left(k u_{\mathrm{A}}^{2-L_f} + (1-k)\, u_0^{2-L_f} \right)^{\frac{2}{2-L_f}} \right]^{\frac{L-L_f}{2}}. \tag{46}$$

The ordinary differential equation in Equation (46) is separable despite being cumbersome. By separating variables and integrating both sides, we can write time t as a function of u_{A}:

$$t = \int_{u_0}^{u_{\mathrm{A}}} x^{1-L_f} \left[x^2 + \left(k x^{2-L_f} + (1-k)\, u_0^{2-L_f} \right)^{\frac{2}{2-L_f}} \right]^{\frac{L_f-L}{2}} dx. \tag{47}$$

The time when u_{A} grows to be comparable with $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|$, for instance $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|/2$, is

$$\begin{aligned} t_{\mathrm{A}} &\approx \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} \int_{u_0}^{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|/2} u_{\mathrm{A}}^{1-L_f} \left[u_{\mathrm{A}}^2 + \left(k u_{\mathrm{A}}^{2-L_f} + (1-k)\, u_0^{2-L_f} \right)^{\frac{2}{2-L_f}} \right]^{\frac{L_f-L}{2}} du_{\mathrm{A}} \\ &\approx \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} u_0^{2-L} \int_1^{\infty} x^{1-L} \left[1 + \left(k + (1-k)\, x^{L_f-2} \right)^{\frac{2}{2-L_f}} \right]^{\frac{L_f-L}{2}} dx. \end{aligned} \tag{48}$$

From Equation (48), we find that the condition for modality A to be learned first in deep intermediate and late fusion linear networks is the same as that for two-layer late fuison linear networks:

$$\tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1} u_0^{2-L_f} I(L, L_f) < \tau \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|^{-1} u_0^{2-L_f} I(L, L_f) \quad \Leftrightarrow \quad \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|} \equiv k \in (0, 1), \tag{49}$$

where we use $I(L, L_f)$ to denote the integral

$$I(L, L_f) = \int_1^{\infty} x^{1-L} \left[1 + \left(k + (1-k)\, x^{L_f-2} \right)^{\frac{2}{2-L_f}} \right]^{\frac{L_f-L}{2}} dx\,.$$

E.2.2. When is the Second Modality Learned?

We next compute the time t_{B} when modality B is learned. During time t_{A} to t_{B}, the network is in manifold $\mathcal{M}_{\mathrm{A}}$ where

$$\boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \quad \Rightarrow \quad u_{\mathrm{A}} = u = \left\| \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \right\|^{\frac{1}{L}},\ t \in (t_{\mathrm{A}}, t_{\mathrm{B}}). \tag{50}$$

We plug $u_{\mathrm{A}}(t_{\mathrm{A}})$, which is very large compared to u_0, into Equation (45) and obtain $u_{\mathrm{B}}(t_{\mathrm{A}})$

$$u_{\mathrm{B}}(t_{\mathrm{A}}) = \left[u_{\mathrm{A}}(t_{\mathrm{A}})^{2-L_f} + (1-k) u_0^{2-L_f} \right]^{\frac{1}{2-L_f}} \approx (1-k)^{\frac{1}{2-L_f}}\, u_0. \tag{51}$$

We then look into the dynamics during the unimodal phase from t_{A} to t_{B}. Substituting $\boldsymbol{W}_{\mathrm{A}}^{\mathrm{tot}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1}$ into Equation (3b), we get

$$\tau \dot{\boldsymbol{W}}_{\mathrm{B}}^{l} = \left(\prod_{j=L_f+1}^{L} \boldsymbol{W}^j \prod_{i=l+1}^{L_f} \boldsymbol{W}_{\mathrm{B}}^i \right)^{\top} \widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}} \left(\prod_{i=1}^{l-1} \boldsymbol{W}_{\mathrm{B}}^i \right)^{\top},\ 1 \leq l \leq L_f, \tag{52}$$

where $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}} = \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \boldsymbol{\Sigma}_{\mathrm{AB}}$. The first-layer weights $\boldsymbol{W}_{\mathrm{B}}^1$ rapidly rotate from $\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}$ to $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}$ at time t_{A} and continues to evolve along $\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}$ during t_{A} to t_{B}. Through the same manner of changing variables, we obtain an ordinary differential equation for u_{B} during t_{A} to t_{B}:

$$\begin{aligned} \tau \dot{u}_{\mathrm{B}} &= \left\| \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \right\|^{1-\frac{L_f}{L}} \|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\| u_{\mathrm{B}}^{L_f-1} \\ \Rightarrow \quad t - t_{\mathrm{A}} &= \tau \left\| \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \right\|^{\frac{L_f}{L}-1} \|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\|^{-1} \frac{u_{\mathrm{B}}^{2-L_f} - u_{\mathrm{B}}(t_{\mathrm{A}})^{2-L_f}}{2-L_f},\ t \in (t_{\mathrm{A}}, t_{\mathrm{B}}). \end{aligned} \tag{53}$$

Plugging in $u_{\mathrm{B}}(t_{\mathrm{A}})$ obtained in Equation (51), we get the time t_{B}:

$$\begin{aligned} t_{\mathrm{B}} &= t_{\mathrm{A}} + \tau \frac{u_{\mathrm{B}}(t_{\mathrm{B}})^{2-L_f} - u_{\mathrm{B}}(t_{\mathrm{A}})^{2-L_f}}{(2-L_f)\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{1-\frac{L_f}{L}}\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\|} \\ &\approx t_{\mathrm{A}} + \tau \frac{-u_{\mathrm{B}}(t_{\mathrm{A}})^{2-L_f}}{(2-L_f)\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{1-\frac{L_f}{L}}\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\|} \\ &\approx t_{\mathrm{A}} + \tau \frac{1-\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1}\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{(L_f-2)\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{1-\frac{L_f}{L}}\|\widetilde{\boldsymbol{\Sigma}}_{y\boldsymbol{x}_{\mathrm{B}}}\|} u_0^{2-L_f}. \end{aligned} \tag{54}$$

Dividing Equation (54) by Equation (48), we arrive at the time ratio Equation (12) in the main text. For intermediate and late fusion linear networks $2 < L_f \leq L$, the time ratio is

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} = 1 + \frac{(\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|)u_0^{L-L_f}}{(L_f-2)\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{1-\frac{L_f}{L}}\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|} I(L, L_f)^{-1},$$

where the integral $I(L, L_f)$ has been defined in Equation (13).

E.3. Time Ratio for $L_f = 2$

When the fusion layer is the second layer $L_f = 2$, the equality in Equation (45) takes following form:

$$\frac{du_{\mathrm{A}}}{du_{\mathrm{B}}} = \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|u_{\mathrm{A}}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|u_{\mathrm{B}}} \quad \Rightarrow \quad \frac{\ln u_{\mathrm{A}} - \ln u_0}{\ln u_{\mathrm{B}} - \ln u_0} = \frac{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}. \tag{55}$$

Consequently, fusion at the second layer is a special case with slightly different expressions for the times. We follow the same procedure as Appendix E.2 and obtain the times for $2 = L_f < L$:

$$t_{\mathrm{A}} \approx \tau \frac{u_0^{2-L}}{\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|} \int_1^{\infty} x^{-1}\left(x^2 + x^{2k}\right)^{1-\frac{L}{2}} dx, \tag{56a}$$

$$t_{\mathrm{B}} \approx t_{\mathrm{A}} + \tau \frac{1-\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\|^{-1}\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|}\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{\frac{2}{L}-1}\ln\frac{1}{u_0}. \tag{56b}$$

The time ratio is

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} = 1 + \frac{(\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| - \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|)u_0^{L-2}\ln\frac{1}{u_0}}{(L_f-2)\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|\left\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}{\boldsymbol{\Sigma}_{\mathrm{A}}}^{-1}\right\|^{1-\frac{2}{L}}} I(L, 2)^{-1}, \tag{57}$$

where the integral is given by

$$I(L, 2) = \int_1^{\infty} x^{-1}\left(x^2 + x^{2k}\right)^{1-\frac{L}{2}} dx. \tag{58}$$

E.4. Time Ratio for Unequal Depth

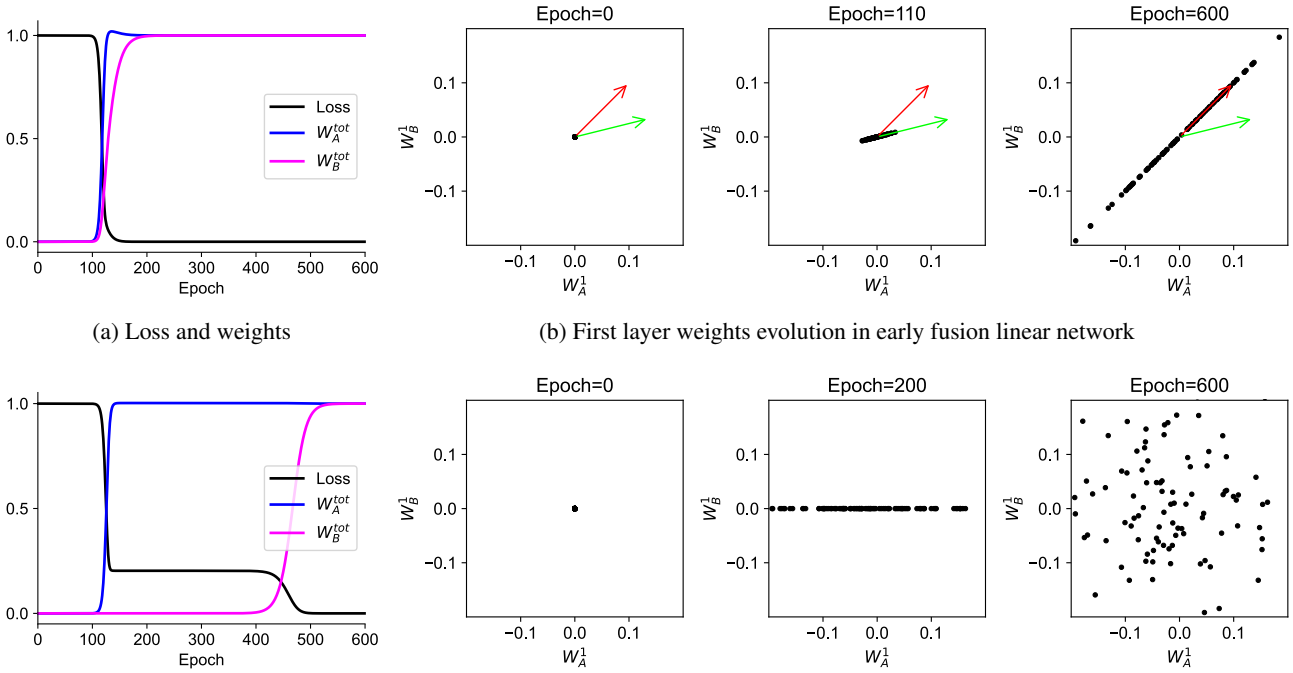
Our time ratio calculations can be carried out for intermediate fusion linear networks with unequal depth between modalities. Consider a intermediate fusion linear network with L_c post-fusion layers, L_{A} pre-fusion layers for the modality A branch, and L_{B} pre-fusion layers for the modality B branch. Assuming $L_{\mathrm{A}}, L_{\mathrm{B}} > 2$ and modality A is learned first, the time ratio is

$$\frac{t_{\mathrm{B}}}{t_{\mathrm{A}}} = 1 + \frac{\frac{u_0^{L_c+L_{\mathrm{A}}-L_{\mathrm{B}}}}{L_{\mathrm{B}}-2}\|\boldsymbol{\Sigma}_{yx_{\mathrm{A}}}\| - \frac{u_0^{L_c}}{L_{\mathrm{A}}-2}\|\boldsymbol{\Sigma}_{yx_{\mathrm{B}}}\|}{\left\|\boldsymbol{\Sigma}_{yx_{\mathrm{A}}}\boldsymbol{\Sigma}_{\mathrm{A}}^{-1}\right\|^{\frac{L_c}{L_{\mathrm{A}}+L_c}}\left\|\boldsymbol{\Sigma}_{yx_{\mathrm{B}}} - \boldsymbol{\Sigma}_{yx_{\mathrm{A}}}\boldsymbol{\Sigma}_{\mathrm{A}}^{-1}\boldsymbol{\Sigma}_{\mathrm{AB}}\right\|} I(L_{\mathrm{A}}, L_{\mathrm{B}}, L_c)^{-1}, \tag{59}$$

where the integral is given by

$$I(L_{\mathrm{A}}, L_{\mathrm{B}}, L_c) = \int_1^{\infty} x^{1-L_{\mathrm{A}}}\left[x^2 + \left(\frac{(L_{\mathrm{B}}-2)\|\boldsymbol{\Sigma}_{yx_{\mathrm{B}}}\|}{(L_{\mathrm{A}}-2)\|\boldsymbol{\Sigma}_{yx_{\mathrm{A}}}\|}u_0^{L_{\mathrm{B}}-L_{\mathrm{A}}}\left(x^{2-L_{\mathrm{A}}}-1\right)+1\right)^{\frac{2}{2-L_{\mathrm{B}}}}\right]^{-\frac{L_c}{2}} dx. \tag{60}$$

As a sanity check, setting $L_{\mathrm{A}} = L_{\mathrm{B}}$ gives us back the same expression as Equation (12).



(a) Loss and weights

(b) First layer weights evolution in early fusion linear network

(c) Loss and weights

(d) First layer weights evolution in late fusion linear network

Figure 10. Feature evolution in two-layer early fusion and late fusion linear networks. We plot the feature evolution, which are first-layer weights in linear networks, corresponding to dynamics in Figures 2b and 2e. (a) Same as Figure 2b. (b) First-layer weights at initialization, during training, and at convergence in panel a. The green arrow denotes the direction of $\boldsymbol{\Sigma}_{y\boldsymbol{x}}$. The red arrow denotes the direction of $\boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}$. (c) Same as Figure 2e. (d) First-layer weights at initialization, during training, and at convergence in panel c. Complementary videos can be found on our website.

F. Feature Evolution in Multimodal Linear Networks

We compare the feature evolution in two-layer early fusion and late fusion linear networks to gain a more detailed understanding of their different learning dynamics. As studied by Atanasov et al. (2022), features of linear networks lie in the first-layer weights. We thus plot the first-layer weights in multimodal linear networks at different times in training to inspect the feature evolution.

F.1. Feature Evolution in Early Fusion Linear Networks

In early fusion linear networks with small initialization, the balancing property $\boldsymbol{W}^1{\boldsymbol{W}^1}^\top = {\boldsymbol{W}^2}^\top\boldsymbol{W}^2$ holds true throughout training, which implies $\boldsymbol{W}^1$ is rank-one throughout training (Du et al., 2018; Ji & Telgarsky, 2019). Specifically, $\boldsymbol{W}^1$ initially aligns with $\boldsymbol{\Sigma}_{y\boldsymbol{x}}$ during the plateau and eventually aligns with $\boldsymbol{\Sigma}_{y\boldsymbol{x}}\boldsymbol{\Sigma}^{-1}$ after $\boldsymbol{W}^1$ have grown in scale and rotated during the brief transition stage. We illustrate this process with Figure 10b. In deep early fusion linear networks, $\boldsymbol{W}^1$ behaves qualitatively the same.

F.2. Feature Evolution in Intermediate and Late Fusion Linear Networks

In intermediate and late fusion linear networks, the balancing property takes a different form as given in Equations (38) and (39). Thus the first-layer weights are not constrained to be rank-one. Specifically, $\boldsymbol{W}_{\text{A}}^1$ grows during the first transition stage while $\boldsymbol{W}_{\text{B}}^1$ remains close to small initialization. After a unimodal phase, $\boldsymbol{W}_{\text{B}}^1$ starts to grow during the second transition stage while $\boldsymbol{W}_{\text{A}}^1$ stays unchanged, shrinks in scale, or expands in scale depending on modality B has zero, positive, or negative correlation with modality A. We illustrate this process with Figure 10d. In deep late and intermediate fusion linear networks, $\boldsymbol{W}^1$ behaves qualitatively the same as in two-layer late fusion linear networks.

G. Superficial Modality Preference

We impose that modality A contributes less to the output, meaning the loss would decrease more if the network visited a saddle in $\mathcal{M}_{\mathrm{B}}$ instead of $\mathcal{M}_{\mathrm{A}}$ as illustrated in Figure 4a:

$$\mathcal{L}(\mathcal{M}_{\mathrm{A}}) > \mathcal{L}(\mathcal{M}_{\mathrm{B}}) \tag{61}$$

Substituting in the saddles defined in Equation (7), we expand and simplify the two mean square losses:

$$\begin{aligned}
\mathcal{L}(\mathcal{M}_{\mathrm{A}}) &= \left\langle (y - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \boldsymbol{x}_{\mathrm{A}})^2 \right\rangle \\
&= \left\langle y^2 \right\rangle - 2\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \langle y\boldsymbol{x}_{\mathrm{A}} \rangle + \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \left\langle \boldsymbol{x}_{\mathrm{A}} \boldsymbol{x}_{\mathrm{A}}^{\top} \right\rangle \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}^{\top} \\
&= \left\langle y^2 \right\rangle - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}^{\top}, \qquad (62a) \\
\mathcal{L}(\mathcal{M}_{\mathrm{B}}) &= \left\langle (y - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1} \boldsymbol{x}_{\mathrm{B}})^2 \right\rangle \\
&= \left\langle y^2 \right\rangle - 2\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \langle y\boldsymbol{x}_{\mathrm{B}} \rangle + \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1} \left\langle \boldsymbol{x}_{\mathrm{B}} \boldsymbol{x}_{\mathrm{B}}^{\top} \right\rangle \boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}^{\top} \\
&= \left\langle y^2 \right\rangle - \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}^{\top}. \qquad (62b)
\end{aligned}$$

Plugging Equation (62) into Equation (61) gives us

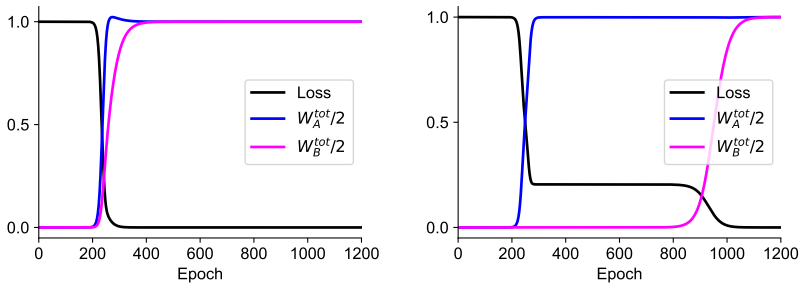
$$\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}} \boldsymbol{\Sigma}_{\mathrm{A}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}^{\top} < \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}} \boldsymbol{\Sigma}_{\mathrm{B}}{}^{-1} \boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}^{\top}.$$

Since we also assume modality A is learned first, thus $\|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{A}}}\| > \|\boldsymbol{\Sigma}_{y\boldsymbol{x}_{\mathrm{B}}}\|$. We thereby arrive at the two inequality conditions Equation (10) in the main text.

H. Two-Layer Multimodal ReLU Networks

H.1. ReLU Networks with Linear Task

Two-layer ReLU networks are trained to learn the same linear task introduced in Figure 2. The loss and weights trajectories in Figure 11 are qualitatively the same as Figure 2, except that learning evolves about two times slower and the converged total weights are two times larger.



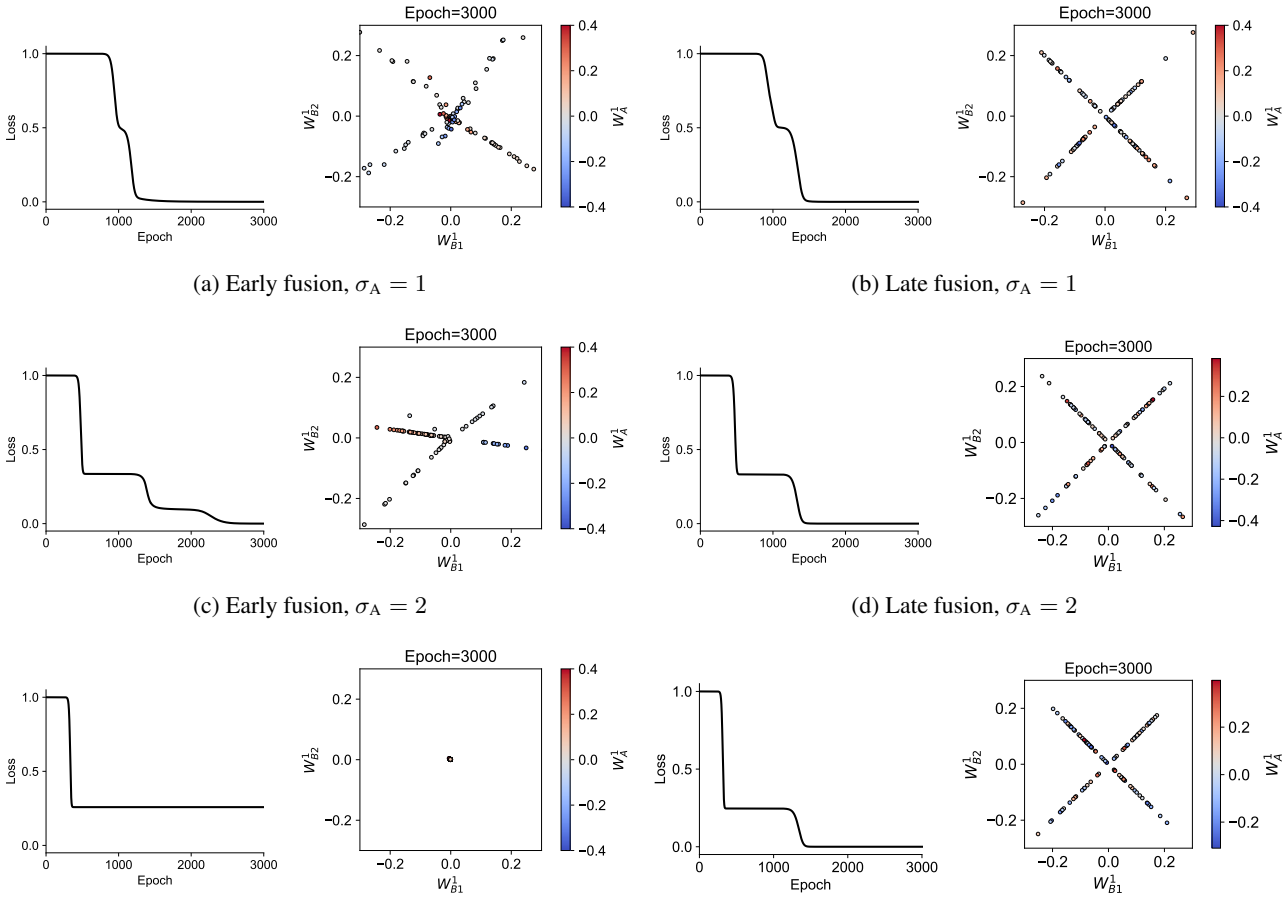
(a) Early fusion (b) Late fusion

Figure 11. Two-layer early fusion and late fusion ReLU networks trained on a linear task. The setting is the same as Figure 2, except that ReLU activation is added to the hidden layer. (a,b) Time trajectories of loss and total weights in the two-layer early fusion and late fusion ReLU network.

H.2. ReLU Networks with Nonlinear Task

We consider a simple nonlinear task of learning $y = \boldsymbol{x}_{\mathrm{A}} + \mathrm{XOR}(\boldsymbol{x}_{\mathrm{B}})$, where $\boldsymbol{x}_{\mathrm{A}} \in \mathbb{R}, \boldsymbol{x}_{\mathrm{B}} \in \{[1,1],[1,-1],[-1,1],[-1,-1]\}$. The term $\mathrm{XOR}(\boldsymbol{x}_{\mathrm{B}})$ refers to performing exclusive-or to the two dimensions of $\boldsymbol{x}_{\mathrm{B}}$. We train two-layer early fusion and late fusion ReLU networks to learn this task with variance of the linear modality $\mathrm{Var}(\boldsymbol{x}_{\mathrm{A}}) \equiv \sigma_{\mathrm{A}} = 1, 2$ or 3. We inspect the features in ReLU networks by plotting the first-layer weights since the features in a two-layer ReLU network lie in its first-layer weights (Xie et al., 2017).

As shown in Figures 12b, 12d and 12f, two-layer late fusion ReLU networks always solve the task by consistently forming the four perpendicular XOR features. We can see two transitions in the loss trajectories of late fusion ReLU networks,



(a) Early fusion, $\sigma_{\mathrm{A}} = 1$

(b) Late fusion, $\sigma_{\mathrm{A}} = 1$

(c) Early fusion, $\sigma_{\mathrm{A}} = 2$

(d) Late fusion, $\sigma_{\mathrm{A}} = 2$

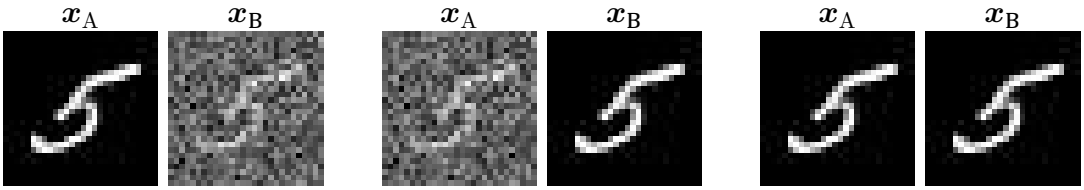
(e) Early fusion, $\sigma_{\mathrm{A}} = 3$

(f) Late fusion, $\sigma_{\mathrm{A}} = 3$

Figure 12. Two-layer early fusion and late fusion ReLU networks trained on an XOR and linear task. The early fusion network has 100 hidden neurons. The late fusion network has 100 hidden neurons in both branches. In every panel, the loss trajectory is plotted on the left and the first-layer weights at the end of training are plotted on the right. The first-layer weights are three-dimensional: the two dimensions of $\boldsymbol{W}^1_{\mathrm{B}}$ are plotted as the position of the dots and $\boldsymbol{W}^1_{\mathrm{A}}$ is plotted as the color of the dots. (a,c,e) Two-layer early fusion ReLU networks. (b,d,f) Two-layer late fusion ReLU networks. Complementary videos can be found on our website.

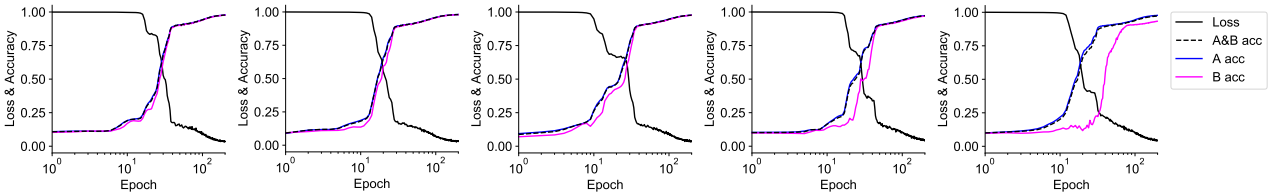
which are similar to the transitions in linear networks. Late fusion ReLU newtorks learn the XOR modality during one transition stage and learn the linear modality during the other. As an example of converged features shown in Figure 12d, $\boldsymbol{W}^1_{\mathrm{B}}$ has taken on the rank-two XOR structure and $\boldsymbol{W}^1_{\mathrm{A}}$ has grown in scale while preserving its independence from $\boldsymbol{W}^1_{\mathrm{B}}$ at initialization.

As shown in Figures 12a, 12c and 12e, two-layer early fusion ReLU networks struggle to extract the XOR features. In the early stage of training, features in $\boldsymbol{W}^1$ favor particular directions in the three-dimensional space that are most correlated with the target output. In comparison with features in late fusion ReLU networks, the first layer weights for modality A do not preserve its independence from weights for modality B as shown in Figure 12c. The larger the variance of the linear modality, the closer the favorable direction is to the direction of the linear modality. In later stage of training, features in $\boldsymbol{W}^1$ can rotate or scatter, giving rise to multiple transition stages as shown in Figure 12c. For a large variance in the linear modality, the features are closely aligned with the linear modality direction and the network can be stuck in a local minimum, failing to learn from the XOR modality as in Figure 12e.



(a) Corrupt $\boldsymbol{x}_{\mathrm{B}}$ (prob 0.6) (b) Corrupt $\boldsymbol{x}_{\mathrm{A}}$ (prob 0.2) (c) Uncorrupted (prob 0.2)

Figure 13. Noisy MNIST dataset.



(a) CNN $L_f = 1$ (b) CNN $L_f = 2$ (c) CNN $L_f = 3$ (d) CNN $L_f = 4$ (e) CNN $L_f = 5$

Figure 14. Loss and accuracy trajectories of multimodal convolutional networks. Six-layer ReLU newtorks with different fusion layer depths are trained on the noisy MNIST dataset shown in Figure 13. The solid black curve plots the training loss. The dotted black curve plots the multimodal classification accuracy, for which both branches are presented with an uncorrupted testset image. The blue curve plots the unimodal classification accuracy of modality A, for which branch A is presented with an uncorrupted testset image and branch B with a blank image. Similarly, for modality B (purple curve). The trajectories are averaged over five random seeds.

I. Noisy MNIST Classification

We perform experiments to validate that our theoretical results, derived for multimodal linear networks, can extend to more realistic scenarios. We train deep fully-connected ReLU networks and deep convolutional networks with different fusion layer depths on a noisy MNIST digit classification task. The results qualitatively align with our conclusion that the deeper the layer at which fusion occurs, the longer it takes to learn from the weaker modality.

In the noisy MNIST classification, the multimodal network receives two written digit images as two modalities, one of which may be corrupted by Gaussian noise. With probability 0.6, modality B is corrupted as in Figure 13a. With probability 0.2, modality A is corrupted as in Figure 13b. With probability 0.2, both modalities are uncorrupted as in Figure 13c. This is a common scenario in multimodal learning where the dominating modality varies per sample.

The deep fully-connected ReLU networks and deep convolutional networks are trained with SGD with cross-entropy loss on the noisy MNIST dataset. The batch size is 1000. The learning rate at the beginning of training is 0.04 for the fully-connected ReLU networks and 0.002 for the convolutional networks. We use a learning rate scheduler that decays the learning rate by 0.996 every epoch. Pytorch's default initialization is used.

The fully-connected ReLU networks have depth $L = 5$ and fusion layer $L_f = 1, 2, \cdots, 5$. The architecture is the same as multimodal deep linear networks except for the ReLU activation. The network width is 500 except for the first layer.

The convolutional networks have six layers: the first five layers are convolutional layers with ReLU activation and the last layer is a fully-connected layer. At a convolutional fusion layer, two inputs are concatenated along the channel and then passed to post-fusion layers. The number of output channels is 32; the kernal size is 3; the stride is 1.

Similar to what we have done with multimodal linear networks, we record the time ratio for the noisy MNIST experiments. The time ratio is computed by dividing the time when the unimodal accuracy of the firstly learned modality reaches 50% by that of the second modality. The time ratio serves as a metric for the relative speed at which the network learns from the second modality compared to the first. All experiments are repeated five times from five random seeds. We report the average and the standard deviation in Table 1.

J. Implementation Details

We provide our code at https://github.com/yedizhang/unimodal-bias.

J.1. Multimodal Linear Networks

Early fusion networks have 100 neurons in every layer. Late fusion networks have 100 neurons in every layer for both branches. Intermediate fusion networks have 100 neurons in every pre-fusion layer for both branches and 100 neurons in every post-fusion layer. All networks are trained with full batch gradient descent with learning rate 0.04. Thus the time constant τ, which appeared in many differential equations such as Equations (3) and (4), is $\tau = 1/0.04 = 25$ for our experiments.

For Figures 3c, 3d, 6b to 6d, 7 and 14 and Table 1, we run experiments with five different random seeds and report the average.

J.2. Data Generation

For data generation, we sample input data points from zero mean normal distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and generate the corresponding target output from a groundtruth linear map $y = \boldsymbol{w}^* \boldsymbol{x}$. All datasets do not contain noise except Figure 5. The number of training samples is 8192 for all experiments except Figure 5. Hence, all experiments, except Figure 5, fall into the underparameterized regime where the training loss well reflects the generalization error.

Note that we do not lose generality by using linear datasets because the learning dyanmics of linear networks as given in Equations (3) and (4) only concern the input correlation matrix $\boldsymbol{\Sigma}$ and input-output correlation matrix $\boldsymbol{\Sigma}_{y\boldsymbol{x}}$. Hence, datasets generated from any distribution and target map with the same $\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_{y\boldsymbol{x}}$ will have the same learning dynamics in linear networks.

J.3. Specifications for Each Figure

Figure 2. Inputs $\boldsymbol{x}_{\mathrm{A}}, \boldsymbol{x}_{\mathrm{B}}$ are scalars with covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(4, 1)$. The target output is generated as $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$. All weights in the early fusion network and the late fusion network are initialized with independent random samples from $\mathcal{N}(0, 10^{-9})$.

Figure 3. Inputs $\boldsymbol{x}_{\mathrm{A}}, \boldsymbol{x}_{\mathrm{B}}$ are scalars with different covariance matrices parameterized as $\boldsymbol{\Sigma} = [\sigma_{\mathrm{A}}^2, \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}; \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}, \sigma_{\mathrm{B}}^2]$, that is $\sigma_{\mathrm{A}}, \sigma_{\mathrm{B}}$ are the variances of modality A,B and ρ is their correlation coefficient. The target output is generated as $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$. All weights are initialized with independent random samples from $\mathcal{N}(0, 10^{-9})$. In this two-dimensional case, the derived time ratio reduces to $1 + (\frac{\sigma_{\mathrm{A}}^2}{\sigma_{\mathrm{B}}^2} - 1)/(1 - \rho^2)$; the amount of mis-attribution reduces to $\rho\frac{\sigma_{\mathrm{B}}}{\sigma_{\mathrm{A}}}$.

Figure 4. In panel (a), the input covariance matrix is $\boldsymbol{\Sigma} = \mathrm{diag}(9, 1)$ and the target output is $y = \boldsymbol{x}_{\mathrm{A}} + 4\boldsymbol{x}_{\mathrm{B}}$. The dotted gray line is below the dotted black, meaning modality B contributes more to the output. The prioritized modality is therefore not the modality that contributes more to the output. In panel (b), the input covariance matrix is $\boldsymbol{\Sigma} = \mathrm{diag}(16, 1)$ and the target output is $y = \boldsymbol{x}_{\mathrm{A}} + 3\boldsymbol{x}_{\mathrm{B}}$. The dotted black line is below the dotted gray, meaning modality A contributes more to the output. The prioritized modality is the modality that contributes more to the output. All weights are initialized with independent random samples from $\mathcal{N}(0, 10^{-9})$.

Figure 5. The inputs $\boldsymbol{x}_{\mathrm{A}}, \boldsymbol{x}_{\mathrm{B}} \in \mathbb{R}^{50}$ have covariance $\boldsymbol{\Sigma}_{\mathrm{A}} = \boldsymbol{I}, \boldsymbol{\Sigma}_{\mathrm{B}} = 3\boldsymbol{I}, \boldsymbol{\Sigma}_{\mathrm{AB}} = \boldsymbol{0}$. The target output is generated as $y = \boldsymbol{w}^* \boldsymbol{x} + \epsilon$, where $\boldsymbol{w}^* = \boldsymbol{1}/10$ and the noise is independently sampled from $\epsilon \sim \mathcal{N}(0, 0.5^2)$. All weights are initialized with independent random samples from $\mathcal{N}(0, 10^{-9})$.

Figure 6. Inputs $\boldsymbol{x}_{\mathrm{A}}, \boldsymbol{x}_{\mathrm{B}}$ are scalars with different covariance matrices parameterized as $\boldsymbol{\Sigma} = [\sigma_{\mathrm{A}}^2, \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}; \rho\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}, \sigma_{\mathrm{B}}^2]$. The target output is generated as $y = \boldsymbol{x}_{\mathrm{A}} + \boldsymbol{x}_{\mathrm{B}}$. Except the initialization scale sweep in Figure 6d, pre-fusion layer weights are initialized with independent random samples from $\mathcal{N}(0, 10^{-4})$; post-fusion layer weights are initialized with independent random samples from $\mathcal{N}(0, 2 \times 10^{-4})$.