Gating is Weighting: Understanding Gated Linear Attention through In-context Learning

Yingcong Li*
University of Michigan
yingcong@umich.edu

Davoud Ataee Tarzanagh*
Samsung SDS Research America
d.tarzanagh@samsung.com

Ankit Singh Rawat
Google Research NYC
ankitsrawat@google.com

Maryam Fazel
University of Washington
mfazel@uw.edu

Samet Oymak
University of Michigan
oymak@umich.edu

Abstract

Linear attention methods offer a compelling alternative to softmax attention due to their efficiency in recurrent decoding. Recent research has focused on enhancing standard linear attention by incorporating gating while retaining its computational benefits. Such Gated Linear Attention (GLA) architectures include highly competitive models such as Mamba and RWKV. In this work, we investigate the in-context learning capabilities of the GLA model and make the following contributions. We show that a multilayer GLA can implement a general class of Weighted Preconditioned Gradient Descent (WPGD) algorithms with data-dependent weights. These weights are induced by the gating mechanism and the input, enabling the model to control the contribution of individual tokens to prediction. To further understand the mechanics of this weighting, we introduce a novel data model with multitask prompts and characterize the optimization landscape of learning a WPGD algorithm. We identify mild conditions under which there exists a unique global minimum, up to scaling invariance, and the associated WPGD algorithm is unique as well. Finally, we translate these findings to explore the optimization landscape of GLA and shed light on how gating facilitates context-aware learning and when it is provably better than vanilla linear attention.

1 Introduction

The Transformer (Vaswani, 2017) has become the de facto standard for language modeling tasks. The key component of the Transformer is the self-attention mechanism, which computes softmax-based similarities between all token pairs. Despite its success, the self-attention mechanism has quadratic complexity with respect to sequence length, making it computationally expensive for long sequences. To address this issue, a growing body of work has proposed near-linear time approaches to sequence modeling. The initial approaches included linear attention and state-space models, both achieving $O(1)$ inference complexity per generated token, thanks to their recurrent form. While these initial architectures typically do not match softmax attention in performance, recent recurrent models such as Mamba (Gu & Dao, 2023; Dao & Gu, 2024), mLSTM (Beck et al., 2024), DeltaNet (Yang et al., 2024b;a), GLA Transformer (Yang et al., 2023), and RWKV-6 (Peng et al., 2024) achieve highly competitive results with the softmax Transformer. Notably, as highlighted in Yang et al. (2023), these architectures can be viewed as variants of *gated linear attention* (GLA), which incorporates a gating mechanism within the recurrence of linear attention. Additionally, Behrouz et al. (2025a;b) unify those models as associative memory modules that optimize internal objectives through iterative algorithms, revealing connections to online optimization and memory management dynamics.

*Equal contribution.

Given a sequence of tokens $(z_i)_{i=1}^{n+1} \in \mathbb{R}^{d+1}$ and the associated query, key, and value embeddings $(\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i)_{i=1}^{n+1} \subset \mathbb{R}^{d+1}$, with $d+1$ being the embedding dimension[1], the GLA recurrence is given by

$$\boldsymbol{S}_i = \boldsymbol{G}_i \odot \boldsymbol{S}_{i-1} + \boldsymbol{v}_i \boldsymbol{k}_i^\top, \quad \text{and} \quad \boldsymbol{o}_i = \boldsymbol{S}_i \boldsymbol{q}_i, \quad i \in \{1, \dots, n+1\}. \tag{GLA}$$

Here, $\boldsymbol{S}_i \in \mathbb{R}^{(d+1)\times(d+1)}$ represents the *2D state variable* with $\boldsymbol{S}_0 = \boldsymbol{0}$; $\boldsymbol{o}_i \in \mathbb{R}^{d+1}$ represents the ith output token; and the *gating variable* $\boldsymbol{G}_i := G(z_i) \in \mathbb{R}^{(d+1)\times(d+1)}$ is applied to the state $\boldsymbol{S}_{i-1}$ through the Hadamard product $\odot$ and G represents the gating function. When $\boldsymbol{G}_i$ is a matrix of all ones, (GLA) reduces to causal linear attention (Katharopoulos et al., 2020).

The central objective of this work is to enhance the mathematical understanding of the GLA mechanism. In-context learning (ICL), one of the most remarkable features of modern sequence models, provides a powerful framework to achieve this aim. ICL refers to the ability of a sequence model to implicitly infer functional relationships from the demonstrations provided in its context window (Brown, 2020; Min et al., 2022). It is inherently related to the model's ability to emulate learning algorithms. Notably, ICL has been a major topic of empirical and theoretical interest in recent years. For example, Ali et al. (2024) show that Mamba embeds implicit attention matrices via data-controlled linear operators, while Sieber et al. (2024) unify SSMs, attention, and RNNs within a dynamical-systems framework. More specifically, a series of works have examined the approximation and optimization characteristics of linear attention, and have provably connected linear attention to the preconditioned gradient descent algorithm (Von Oswald et al., 2023; Ahn et al., 2024; Zhang et al., 2024). Given that the GLA recurrence in (GLA) has a richer design space, this leads us to ask:

What learning algorithm does GLA emulate in ICL?

Contributions: The (GLA) recurrence enables the sequence model to weight past information in a data-dependent manner through the gating mechanism $(\boldsymbol{G}_i)_{i=1}^n$. Building on this observation, we demonstrate that a GLA model can implement a *data-dependent Weighted Preconditioned Gradient Descent (WPGD)* algorithm. Specifically, a one-step WPGD with scalar gating, where all entries of $\boldsymbol{G}_i$ are identical, is described by the prediction:

$$\hat{y} = \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{X}^\top (\boldsymbol{y} \odot \boldsymbol{\omega}). \tag{1}$$

Here, $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is the input feature matrix; $\boldsymbol{y} \in \mathbb{R}^n$ is the associated label vector; $\boldsymbol{x} \in \mathbb{R}^d$ represents the test/query input to predict; $\boldsymbol{P} \in \mathbb{R}^{d\times d}$ is the preconditioning matrix; and $\boldsymbol{\omega} \in \mathbb{R}^n$ weights the individual samples. When $\boldsymbol{\omega}$ is fixed, we drop "data-dependent" and simply refer to this algorithm as the WPGD algorithm. However, for GLA, $\boldsymbol{\omega}$ depends on the data through recursive multiplication of the gating variables. Building on this formalism, we make the following specific contributions:

- ◇ **ICL capabilities of GLA (§3):** Through constructive arguments, we demonstrate that a multilayer GLA model can implement data-dependent WPGD iterations, with weights induced by the gating function. This construction sheds light on the role of causal masking and the expressivity distinctions between scalar- and vector-valued gating functions.
- ◇ **Landscape of one-step WPGD (§4):** The GLA⇔WPGD connection motivates us to ask: *How does WPGD weigh demonstrations in terms of their relevance to the query?* To address this, we study the fundamental problem of learning an optimal WPGD algorithm: Given a tuple $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}, y) \sim \mathcal{D}$, with $y \in \mathbb{R}$ being the label associated with the query, we investigate the population risk minimization:

$$\begin{aligned} &\mathcal{L}^\star_{\texttt{WPGD}} := \min_{\boldsymbol{P}\in\mathbb{R}^{d\times d}, \boldsymbol{\omega}\in\mathbb{R}^n} \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega}), \\ &\text{where } \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega}) = \mathbb{E}_{\mathcal{D}}\left[\left(y - \boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{X}^\top (\boldsymbol{y} \odot \boldsymbol{\omega})\right)^2\right]. \end{aligned} \tag{2}$$

[1]The sequence length and embedding dimension are set to $n+1$ and $d+1$ per the prompt definition in (3).

As our primary mathematical contribution, we characterize the loss landscape under a general multitask data setting, where the tasks associated with the demonstrations $(\boldsymbol{X}, \boldsymbol{y})$ have varying degrees of correlation to the target task $(\boldsymbol{x}, y)$. We carefully analyze this loss landscape and show that, under mild conditions, there is a unique global minimum $(\boldsymbol{P}, \boldsymbol{\omega})$ up to scaling invariance, and the associated WPGD algorithm is also unique.

- **Loss landscape of one-layer GLA (§5):** The landscape is highly intricate due to the recursively multiplied gating variables. We show that learning the optimal GLA layer can be connected to solving (2) with a constraint $\boldsymbol{\omega} \in \mathcal{W}$, where the restriction $\mathcal{W}$ is induced by the choice of gating function and input space. Solidifying this connection, we introduce a multitask prompt model under which we characterize the loss landscape of GLA and the influence of task correlations. Our analysis and experiments reveal insightful distinctions between linear attention, GLA with scalar gating, and GLA with vector-valued gating.

2 Problem setup

Notations. $\mathbb{R}^d$ is the d-dimensional real space, with $\mathbb{R}^d_+$ and $\mathbb{R}^d_{++}$ as its positive and strictly positive orthants. The set $[n]$ denotes $\{1, \ldots, n\}$. Bold letters, e.g., $\boldsymbol{a}$ and $\boldsymbol{A}$, represent vectors and matrices. The identity matrix of size n is denoted by $\boldsymbol{I}_n$. The symbols $\mathbf{1}$ and $\mathbf{0}$ denote the all-one and all-zero vectors or matrices of proper size, such as $\mathbf{1}_d \in \mathbb{R}^d$ and $\mathbf{1}_{d\times d} \in \mathbb{R}^{d\times d}$. The subscript is omitted when the dimension is clear from the context. The Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is written as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The Hadamard product (element-wise multiplication) is denoted by $\odot$, and Hadamard division (element-wise division) is denoted by $\oslash$. Given any $\boldsymbol{a}_{i+1}, \ldots, \boldsymbol{a}_j \in \mathbb{R}^d$, we define $\boldsymbol{a}_{i:j} = \boldsymbol{a}_{i+1} \odot \cdots \odot \boldsymbol{a}_j$ for $i < j$, and $\boldsymbol{a}_{i:i} = \mathbf{1}_d$.

The objective of this work is to develop a theoretical understanding of GLA through ICL. The optimization landscape of standard linear attention has been a topic of significant interest in the ICL literature (Ahn et al., 2024; Li et al., 2024b). Following these works, we consider the input prompt

$$\boldsymbol{Z} = \begin{bmatrix} z_1 & \ldots & z_n & z_{n+1} \end{bmatrix}^\top = \begin{bmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_n & \boldsymbol{x}_{n+1} \\ y_1 & \ldots & y_n & 0 \end{bmatrix}^\top \in \mathbb{R}^{(n+1)\times(d+1)}, \tag{3}$$

where tokens encode the input-label pairs $(\boldsymbol{x}_i, y_i)_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$.

We aim to enable ICL by training a sequence model $F : \mathbb{R}^{(n+1)\times(d+1)} \to \mathbb{R}$ that predicts the label $y := y_{n+1}$ associated with the query $\boldsymbol{x} := \boldsymbol{x}_{n+1}$. This model will utilize the demonstrations $(\boldsymbol{x}_i, y_i)_{i=1}^n$ to infer the mapping between $\boldsymbol{x}$ and y. Assuming that the data is distributed as $(y, \boldsymbol{Z}) \sim \mathcal{D}$, the ICL objective is defined as

$$\mathcal{L}(F) = \mathbb{E}_{\mathcal{D}}\left[(y - F(\boldsymbol{Z}))^2\right]. \tag{4}$$

Linear attention and shared-task distribution. Central to our paper is the choice of the function class F. When F is a linear attention model, the prediction $\hat{y}_F := F(\boldsymbol{Z})$ takes the form $\hat{y}_F = z_{n+1}^\top \boldsymbol{W}_q \boldsymbol{W}_k^\top \boldsymbol{Z}^\top \boldsymbol{Z} \boldsymbol{W}_v \boldsymbol{h}$, where $\boldsymbol{W}_k, \boldsymbol{W}_q, \boldsymbol{W}_v \in \mathbb{R}^{(d+1)\times(d+1)}$ are attention parameters, and $\boldsymbol{h} \in \mathbb{R}^{d+1}$ is the linear prediction head.

We assume that the in-context input-label pairs follow a *shared-task distribution*, where $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$, $\boldsymbol{x}_i$ are i.i.d. with $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, and $y_i \sim \mathcal{N}(\boldsymbol{\beta}^\top \boldsymbol{x}_i, \sigma^2)$, where $\sigma \geq 0$ represents the noise level. Under this shared-task distribution, it is shown (Von Oswald et al., 2023; Ahn et al., 2024; Zhang et al., 2024) that the optimal one-layer linear attention prediction coincides with the one-step optimal preconditioned gradient descent (PGD). In particular, considering the data distribution discussed above, we have

$$\hat{y}_F = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}, \quad \text{where} \quad \hat{\boldsymbol{\beta}} = \boldsymbol{P}^\star \boldsymbol{X}^\top \boldsymbol{y}, \tag{5}$$

and

$$\boldsymbol{P}^\star := \underset{\boldsymbol{P}\in\mathbb{R}^{d\times d}}{\operatorname{argmin}}\ \mathbb{E}\left[\left(y - \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}^\top \boldsymbol{y}\right)^2\right] \quad \text{with } \boldsymbol{X} := [\boldsymbol{x}_1\ \cdots\ \boldsymbol{x}_n]^\top \text{ and } \boldsymbol{y} := [y_1\ \cdots\ y_n]^\top. \tag{6}$$

Linear attention and gating. Given the input prompt $\boldsymbol{Z}$ as in (3), let $(\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i) = (\boldsymbol{W}_q^\top \boldsymbol{z}_i, \boldsymbol{W}_k^\top \boldsymbol{z}_i, \boldsymbol{W}_v^\top \boldsymbol{z}_i)$ be the corresponding query, key, and value embeddings. The output of *causal* linear attention at time i can be computed using (GLA) with $\boldsymbol{G}_i = \boldsymbol{1}$. This recurrent form implies that linear attention has $O(d^2)$ cost, that is independent of sequence length n, to generate per-token. GLA follows the same structure as linear attention but with a gating mechanism (i.e., $\boldsymbol{G}_i \neq \boldsymbol{1}$), which equips the model with the option to pass or supress the history. As discussed in Yang et al. (2023), the different choices of the gating function correspond to different popular recurrent architectures such as Mamba (Gu & Dao, 2023), Mamba2 (Dao & Gu, 2024), RWKV (Peng et al., 2024), etc.

We will show that GLA can weigh the context window through gating, thus, its capabilities are linked to the WPGD algorithm described in (8). This will in turn facilitate GLA to effectively learn *multitask prompt distributions* described by $y_i \sim \mathcal{N}(\boldsymbol{\beta}_i^\top \boldsymbol{x}_i, \sigma^2)$ with $\boldsymbol{\beta}_i$'s not necessarily identical.

3 What gradient methods can GLA emulate?

In this section, we investigate the ICL capabilities of GLA and show that under suitable instantiations of model weights, GLA can implement *data-dependent* WPGD.

3.1 GLA as a data-dependent WPGD predictor

Data-Dependent WPGD. Given $\boldsymbol{X}$ and $\boldsymbol{y}$ as defined in (6), consider the weighted least squares objective $\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n \omega_i \cdot (y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i)^2$ with weights $\boldsymbol{\omega} = [\omega_1, \omega_2, \cdots, \omega_n]^\top \in \mathbb{R}^n$. To optimize this, we use gradient descent (GD) starting from zero initialization, $\boldsymbol{\beta}_0 = \boldsymbol{0}$ with a step size of $\eta = 1/2$. One step of standard GD is given by

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 - \eta \nabla \mathcal{L}(\boldsymbol{\beta}_0) = \sum_{i=1}^n \omega_i \cdot \boldsymbol{x}_i y_i = \boldsymbol{X}^\top (\boldsymbol{\omega} \odot \boldsymbol{y}).$$

Given a test/query feature $\boldsymbol{x}$, the corresponding prediction is $\hat{y} = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}$ where $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_1$. Additionally, if we were using PGD with a preconditioning matrix $\boldsymbol{P} \in \mathbb{R}^{d\times d}$, $\boldsymbol{\beta}_0 = \boldsymbol{0}$, and $\eta = 1/2$, then a single iteration results in

$$\hat{y} = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}, \quad \text{where} \quad \hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0 - \eta \boldsymbol{P} \nabla \mathcal{L}(\boldsymbol{\beta}_0) = \boldsymbol{P}\boldsymbol{X}^\top (\boldsymbol{\omega} \odot \boldsymbol{y}). \tag{7}$$

Above is the basic *sample-wise* WPGD predictor which weights individual datapoints, in contrast to the PGD predictor as presented in (5). It turns out, *vector-valued gating* can facilitate a more general estimator which weights individual coordinates. To this aim, we introduce an extension as follows: Let $\boldsymbol{P}_1, \boldsymbol{P}_2 \in \mathbb{R}^{d\times d}$ denote the preconditioning matrices, and let $\boldsymbol{\Omega} \in \mathbb{R}^{n\times d}$ denote the *vector-valued weighting* matrix. Note that $\boldsymbol{\Omega}$ is a weight matrix that enables coordinate-wise weighting. Throughout the paper, we use $\boldsymbol{\omega}$ and $\boldsymbol{\Omega}$ to denote sample-wise and coordinate-wise weighting parameters, corresponding to scalar and vector gating strategies, respectively. Then given $\boldsymbol{\Omega}$, we can similarly define

$$\boldsymbol{\beta}_1^{\mathrm{gd}}(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{\Omega}) := \boldsymbol{P}_2 (\boldsymbol{X}\boldsymbol{P}_1 \odot \boldsymbol{\Omega})^\top \boldsymbol{y} \tag{8a}$$

as one-step of (generalized) WPGD. Its corresponding prediction on a test query $\boldsymbol{x}$ is:

$$\hat{y} = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}, \quad \text{where} \quad \hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_1^{\mathrm{gd}}(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{\Omega}). \tag{8b}$$

We note that by removing the weighting matrix $\boldsymbol{\Omega}$, (8) reduces to standard PGD (cf. (5)) and setting $\boldsymbol{\Omega} = \boldsymbol{\omega}\boldsymbol{1}_d^\top$ reduces to sample-wise WPGD (cf. (7)). Li et al. (2024b) has demonstrated

that H3-like models implement one-step *sample-wise* WPGD, and they focus on the shared-task distribution where $\boldsymbol{\beta}_i \equiv \boldsymbol{\beta}$ for all $i \in [n]$. In contrast, our work considers a more general data setting where tasks within an in-context prompt are not necessarily identical.

We introduce the following model constructions under which we establish the equivalence between GLA (cf. (GLA)) and WPGD (cf. (8)), where the weighting matrix is induced by the input data and the gating function. Inspired by prior works (Von Oswald et al., 2023; Ahn et al., 2024), we consider the following restricted attention matrices:

$$\boldsymbol{W}_k = \begin{bmatrix} \boldsymbol{P}_k & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix}, \quad \boldsymbol{W}_q = \begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{W}_v = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{9}$$

where $\boldsymbol{P}_k, \boldsymbol{P}_q \in \mathbb{R}^{d\times d}$. Note that the $(d+1, d+1)$-th entry of $\boldsymbol{W}_v$ is set to one for simplicity. More generally, this entry can take any nonzero value, e.g., $v \in \mathbb{R}$. Parameterizing $\boldsymbol{W}_q$ with $\boldsymbol{P}_q / v$ produces the same output as in (9).

Theorem 1. *Recall (GLA) with input sequence $\boldsymbol{Z} = [\boldsymbol{z}_1 \ \ldots \ \boldsymbol{z}_n \ \boldsymbol{z}_{n+1}]^\top$ defined in (3), the gating $G(\boldsymbol{z}_i) = \boldsymbol{G}_i \in \mathbb{R}^{(d+1)\times(d+1)}$, and outputs $(\boldsymbol{o}_i)_{i=1}^{n+1}$. Consider model construction in (9) and take the last coordinate of the last token output denoted by $(\boldsymbol{o}_{n+1})_{d+1}$ as a prediction. Then, we have*

$$f_{\mathsf{GLA}}(\boldsymbol{Z}) := (\boldsymbol{o}_{n+1})_{d+1} = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}, \quad \text{where} \quad \hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_1^{gd}(\boldsymbol{P}_k, \boldsymbol{P}_q, \boldsymbol{\Omega}).$$

Here, $\boldsymbol{\beta}_1^{gd}(\cdot)$ is a one-step WPGD feature predictor defined in (8a); $\boldsymbol{P}_k$ and $\boldsymbol{P}_q$ correspond to attention weights following (9); and $\boldsymbol{\Omega} = [\boldsymbol{g}_{1:n+1} \ \cdots \ \boldsymbol{g}_{n:n+1}]^\top \in \mathbb{R}^{n\times d}$, where $\boldsymbol{g}_{i:n+1}, i \in [n]$ is given by

$$\boldsymbol{g}_{i:n+1} := \boldsymbol{g}_{i+1} \odot \boldsymbol{g}_{i+2} \cdots \boldsymbol{g}_{n+1} \in \mathbb{R}^d, \quad \text{and} \quad \boldsymbol{G}_i = \begin{bmatrix} * & * \\ \boldsymbol{g}_i^\top & * \end{bmatrix}. \tag{10}$$

Here and throughout, we use $*$ to fill the entries of the matrices that do not affect the final output. Based on the model construction in (9), these entries can be assigned any value.

Observe that, crucially, since $\boldsymbol{g}_i$ is associated with $\boldsymbol{z}_i$, $\boldsymbol{z}_i$ influences the weighting of $\boldsymbol{z}_j$ for all $j < i$. We defer the proof of Theorem 1 to the Appendix D.1. It is noticeable that only d of the total $(d+1)^2$ entries in each gating matrix $\boldsymbol{G}_i$ are useful due to the model construction presented in (9). However, if we relax the weight restriction, e.g., $\boldsymbol{W}_v = [\boldsymbol{0}_{(d+1)\times d} \ \ \boldsymbol{u}]^\top$, where $\boldsymbol{u} \in \mathbb{R}^{d+1}$, then the weighting matrix $\boldsymbol{\Omega}$ in Theorem 1 is associated with all rows of the $\boldsymbol{G}_i$ matrices. We defer to Eqn. (27) and the discussion in Appendix D.1.

Capabilities of multi-layer GLA. Ahn et al. (2024) demonstrated that, with appropriate construction, an L-layer linear attention model performs L steps of PGD on the dataset $(\mathbf{x}_i, y_i)_{i=1}^n$ provided within the prompt. In Appendix A, we extend this analysis to multi-layer GLA and characterize the algorithmic class it can emulate. Importantly, Ahn et al. (2024) does not account for *causal masking*, which is a fundamental aspect of multi-layer GLA due to its recurrent structure as described in (GLA). Our main result, Theorem 6 in Appendix A, establishes that an L-layer GLA implements L steps of WPGD, where the gradients are computed in a recurrent form.

GLA with scalar gating. Theorem 1 establishes a connection between one-layer GLA (cf. (GLA)) and one-step WPGD (cf. (8)), where the weighting in WPGD corresponds to the gating $G(\boldsymbol{z}_i) = \boldsymbol{G}_i$ in GLA, as detailed in Theorem 1. Now let us consider the widely used types of gating functions, such as $\boldsymbol{G}_i = \boldsymbol{\alpha}_i \mathbf{1}_{d+1}^\top$ (Yang et al., 2023; Katsch, 2023; Qin et al., 2024; Peng et al., 2024) or $\boldsymbol{G}_i = \gamma_i \mathbf{1}_{(d+1),(d+1)}$ (Dao & Gu, 2024; Beck et al., 2024; Peng et al., 2021; Sun et al., 2024), where $\boldsymbol{\alpha}_i \in \mathbb{R}^{d+1}$ and $\gamma_i \in \mathbb{R}$. In both cases, the gating matrices in (10) take the form of $\begin{bmatrix} * & * \\ g_i \mathbf{1}_d^\top & * \end{bmatrix}$ for some $g_i \in \mathbb{R}$, thus simplifying the predictor to a sample-wise WPGD, as given by

$$f_{\mathsf{GLA}}(\boldsymbol{Z}) = \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}, \quad \text{with} \quad \hat{\boldsymbol{\beta}} = \boldsymbol{P}\boldsymbol{X}^\top(\boldsymbol{\omega} \odot \boldsymbol{y}), \tag{11}$$

where $\boldsymbol{P} = \boldsymbol{P}_q \boldsymbol{P}_k^\top$ and $\boldsymbol{\omega} = [g_{1:n+1} \ \cdots \ g_{n:n+1}]^\top \in \mathbb{R}^n$. In the remainder, we will mostly focus on the one-layer GLA with scalar gating as presented in (11).

4 Optimization landscape of WPGD

In this section, we explore the problem of learning the optimal sample-wise WPGD algorithm described in (11), a key step leading to our analysis of GLA. The problem is as follows. Recap from (6) that we are given the tuple $(\boldsymbol{x}, y, \boldsymbol{X}, \boldsymbol{y}) \sim \mathcal{D}$, where $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is the input matrix, $\boldsymbol{y} \in \mathbb{R}^n$ is the label vector, $\boldsymbol{x} \in \mathbb{R}^d$ is the query, and $y \in \mathbb{R}$ is its associated label. The goal is to use $\boldsymbol{X}, \boldsymbol{y}$ to predict y given $\boldsymbol{x}$ via the one-step WPGD prediction $\hat{y} = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}}$, with $\hat{\boldsymbol{\beta}}$ as in (11). The algorithm learning problem is given by (2) which minimizes the WPGD risk $\mathbb{E}_{\mathcal{D}}[\left(y - \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right)^2]$.

Prior research (Mahankali et al., 2023; Li et al., 2024b; Ahn et al., 2024) has studied the problem of learning PGD when input-label pairs follow an i.i.d. distribution. It is worth noting that while Li et al. (2024b) establishes a connection between H3-like models and (11) similar to ours, their work assumes that the optimal $\boldsymbol{\omega}$ consists of all ones and does not specifically explore the optimization landscape of $\boldsymbol{\omega}$ when in-context samples are not generated from the same task vector $\boldsymbol{\beta}$. Departing from this, we introduce a realistic model where each input-label pair is allowed to come from a distinct task.

Definition 1 (Correlated Task Model)**.** *Suppose $\boldsymbol{\beta}_i \in \mathbb{R}^d \sim \mathcal{N}(0, \boldsymbol{I})$ are jointly Gaussian for all $i \in [n+1]$. Define*

$$\boldsymbol{\beta} := \boldsymbol{\beta}_{n+1}, \quad \boldsymbol{B} := [\boldsymbol{\beta}_1 \ \dots \ \boldsymbol{\beta}_n]^\top, \quad \boldsymbol{R} := \frac{1}{d}\mathbb{E}[\boldsymbol{B}\boldsymbol{B}^\top], \quad \text{and} \quad \boldsymbol{r} := \frac{1}{d}\mathbb{E}[\boldsymbol{B}\boldsymbol{\beta}]. \tag{12}$$

Note that in (12), we have $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{r} \in \mathbb{R}^n$, with normalization ensuring that the entries of $\boldsymbol{R}$ and $\boldsymbol{r}$ lie in the range $[-1, 1]$, corresponding to correlation coefficients. Further, due to the joint Gaussian nature of the tasks, for any $i, j \in [n+1]$, the residual $\boldsymbol{\beta}_i - r_{ij}\boldsymbol{\beta}_j$ is independent of $\boldsymbol{\beta}_j$.

Definition 2 (Multitask Distribution)**.** *Let $(\boldsymbol{\beta}_i)_{i=1}^{n+1}$ be drawn according to the correlated task model of Definition 1, $(\boldsymbol{x}_i)_{i=1}^{n+1} \in \mathbb{R}^d$ be i.i.d. with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, and $y_i \sim \mathcal{N}(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i, \sigma^2)$ for all $i \in [n+1]$.*

Definition 3. *Let the eigen decompositions of $\boldsymbol{\Sigma}$ and $\boldsymbol{R}$ be denoted by $\boldsymbol{\Sigma} = \boldsymbol{U}\mathrm{diag}(\boldsymbol{s})\boldsymbol{U}^\top$ and $\boldsymbol{R} = \boldsymbol{E}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{E}^\top$, where $\boldsymbol{s} = [s_1 \ \cdots \ s_d]^\top \in \mathbb{R}^d_{++}$ and $\boldsymbol{\lambda} = [\lambda_1 \ \cdots \ \lambda_n]^\top \in \mathbb{R}^n_+$. Let $s_{\min}$ and $s_{\max}$ denote the smallest and largest eigenvalues of $\boldsymbol{\Sigma}$, respectively. Further, let $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest nonzero eigenvalues of $\boldsymbol{R}$. Define the effective spectral gap of $\boldsymbol{\Sigma}$ and $\boldsymbol{R}$, respectively, as*

$$\Delta_{\boldsymbol{\Sigma}} := s_{\max} - s_{\min}, \quad \text{and} \quad \Delta_{\boldsymbol{R}} := \lambda_{\max} - \lambda_{\min}. \tag{13}$$

Assumption 1. *The correlation vector $\boldsymbol{r}$ from (12) lies in the range of $\boldsymbol{E}$, i.e., $\boldsymbol{r} = \boldsymbol{E}\boldsymbol{a}$ for some nonzero $\boldsymbol{a} = [a_1 \ \cdots \ a_n]^\top \in \mathbb{R}^n$.*

Assumption 1 essentially ensures that $\boldsymbol{r}$ (representing the correlations between in-context tasks) can be expressed as a linear transformation of a vector $\boldsymbol{a}$ with at least one nonzero value. Note that since $\boldsymbol{r}$ lies in the range of the eigenvector matrix $\boldsymbol{E}$, it also lies in the range of $\boldsymbol{R}$ defined in (12). This guarantees that the correlation structure is non-degenerate, meaning that all elements of $\boldsymbol{r}$ are influenced by meaningful correlations. Assumption 1 avoids trivial cases where there are no correlations between tasks. By requiring at least one nonzero element in $\boldsymbol{a}$, the assumption ensures that the tasks are interrelated.

The following theorem characterizes the stationary points $(\boldsymbol{P}, \boldsymbol{\omega})$ of the WPGD objective (2).

Theorem 2. *Consider independent linear data as described in Definition 2. Suppose Assumption 1 on the correlation vector $\boldsymbol{r}$ holds. Define the functions $h_1 : \mathbb{R}_+ \to \mathbb{R}_+$ and $h_2 : [1, \infty) \to \mathbb{R}_+$ as*

$$h_1(\bar{\gamma}) := \left(\sum_{i=1}^n \frac{\lambda_i a_i^2}{(1+\lambda_i\bar{\gamma})^2}\right)\left(\sum_{i=1}^n \frac{a_i^2}{(1+\lambda_i\bar{\gamma})^2}\right)^{-1}, \tag{14a}$$

$$h_2(\gamma) := \left(1 + M\left(\sum_{i=1}^d \frac{s_i^2}{(M+s_i\gamma)^2}\right)\left(\sum_{i=1}^d \frac{s_i^3}{(M+s_i\gamma)^2}\right)^{-1}\right)^{-1}, \tag{14b}$$

where $\{s_i\}_{i=1}^d$ *and* $\{\lambda_i\}_{i=1}^n$ *are the eigenvalues of* $\mathbf{\Sigma}$ *and* $\boldsymbol{R}$*, respectively;* $\{a_i\}_{i=1}^n$ *are given in Assumption 1; and* $M = \sigma^2 + \sum_{i=1}^d s_i$.

The risk function $\mathcal{L}_{\mathtt{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega})$ *in (2) has a stationary point* $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$*, up to rescaling, defined as*

$$\boldsymbol{P}^\star = \mathbf{\Sigma}^{-\frac{1}{2}} \left(\frac{\gamma^\star}{M} \cdot \mathbf{\Sigma} + \boldsymbol{I} \right)^{-1} \mathbf{\Sigma}^{-\frac{1}{2}}, \quad \text{and} \quad \boldsymbol{\omega}^\star = \left(h_2(\gamma^\star) \cdot \boldsymbol{R} + \boldsymbol{I} \right)^{-1} \boldsymbol{r}, \tag{15}$$

where $\gamma^\star$ *is a fixed point of composite function* $h_1(h_2(\gamma)) + 1$.

Theorem 2 characterizes the stationary points $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$, which exist up to re-scaling. This result presents the first landscape analysis of GLA for the joint learning of $(\boldsymbol{P}, \boldsymbol{\omega})$, while also exploring the stationary points $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$. In the following, we provide mild conditions on effective spectral gaps of $\boldsymbol{R}$ and $\mathbf{\Sigma}$ under which a unique (global) minimum $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$ exists.

Theorem 3 (Uniqueness of the WPGD Predictor)**.** *Consider independent linear data as given in Definition 2. Suppose Assumption 1 on the correlation vector* $\boldsymbol{r}$ *holds, and*

$$\Delta_{\mathbf{\Sigma}} \cdot \Delta_{\boldsymbol{R}} < M + s_{\min}, \tag{16}$$

where $\Delta_{\mathbf{\Sigma}}$ *and* $\Delta_{\boldsymbol{R}}$ *denote the effective spectral gaps of* $\mathbf{\Sigma}$ *and* $\boldsymbol{R}$*, respectively, as given in (13);* $s_{\min}$ *is the smallest eigenvalue of* $\mathbf{\Sigma}$*; and* $M = \sigma^2 + \sum_{i=1}^d s_i$.

T1. *The function* $h_1(h_2(\gamma)) + 1$ *is a contraction mapping and admits a unique fixed point* $\gamma = \gamma^\star$.

T2. *The loss* $\mathcal{L}_{\mathtt{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega})$ *has a unique (global) minima* $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$*, up to re-scaling, given by (15).*

Theorem 3 establishes mild conditions under which a unique (global) minimum $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$ exists, up to scaling invariance, and guarantees the uniqueness of the associated WPGD algorithm. It provides the first global landscape analysis for GLA and generalizes prior work (Li et al., 2024b; Ahn et al., 2024) on the global landscape by extending the optimization properties of linear attention to the more complex GLA with joint $(\boldsymbol{P}, \boldsymbol{\omega})$ optimization.

Remark 1. An interesting observation about the optimal gating parameter $\boldsymbol{\omega}^\star$ is its connection to the correlation matrix $\boldsymbol{R}$, which captures the task correlations in a multitask learning setting. Specifically, the optimal gating given in (15) highlights how $\boldsymbol{\omega}^\star$ depends directly on both the task correlation matrix $\boldsymbol{R}$ and the vector $\boldsymbol{r}$, which encodes the correlations between the tasks and the target task.

Remark 2. Condition (16) provides a *sufficient* condition for the uniqueness of a fixed point. This implies that whenever $\Delta_{\mathbf{\Sigma}} \cdot \Delta_{\boldsymbol{R}} < M + s_{\min}$, the mapping $h_1(h_2(\gamma)) + 1$ is a contraction, ensuring the existence of a unique fixed point. However, there may be cases where the mapping $h_1(h_2(\gamma)) + 1$ does not satisfy Condition (16), yet a unique fixed point (and a unique global minimum) still exists. This is because the Banach Fixed-Point Theorem does not provide a *necessary* condition.

Corollary 1. *Suppose* $\mathbf{\Sigma} = \boldsymbol{I}$*. Then,* $\Delta_{\mathbf{\Sigma}} = 0$*, satisfying Condition (16), and we have* $h_2(\gamma^\star) = \frac{1}{d+\sigma^2+1}$*, which yields* $\boldsymbol{P}^\star = \boldsymbol{I}$ *and* $\boldsymbol{\omega}^\star = (\boldsymbol{R} + (d + \sigma^2 + 1)\boldsymbol{I})^{-1}\boldsymbol{r}$*. Thus, the optimal risk* $\mathcal{L}^\star_{\mathtt{WPGD}}$ *defined in (2) is given by*

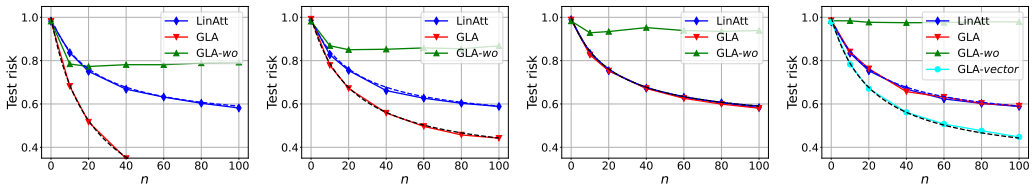
$$\mathcal{L}^\star_{\mathtt{WPGD}} = d + \sigma^2 - d \cdot \boldsymbol{r}^\top \left(\boldsymbol{R} + (d + \sigma^2 + 1)\boldsymbol{I} \right)^{-1} \boldsymbol{r}. \tag{17}$$

5 Optimization landscape of GLA

In this section, we analyze the loss landscape for training a one-layer GLA model and explore the scenarios under which GLA can reach the optimal WPGD risk.

5.1 Multi-task prompt model

Following Definitions 1 and 2, we consider the multi-task prompt setting with K correlated tasks $(\boldsymbol{\beta}_k)_{k=1}^K$ and one query task $\boldsymbol{\beta}$. For each correlated task $k \in [K]$, a prompt of length



(a) $(r_1, r_2) = (0, 1)$ (b) $(r_1, r_2) = (0.2, 0.8)$ (c) $(r_1, r_2) = (0.5, 0.5)$ (d) $(r_1, r_2) = (0.8, 0.2)$

Figure 1: We consider four different types of model training: `LinAtt` (blue solid): Standard linear attention training. `GLA` (red solid): GLA training using prompts with delimiters (see (19)) and scalar gating. `GLA-wo` (green solid): GLA training using prompts without delimiters and with scalar gating. `GLA-vector` (cyan solid): GLA training using prompts with delimiters and vector gating. The blue and black dashed curves represent the optimal linear attention and WPGD risks from (24) and (17), respectively, as the number of in-context examples n increases. Implementation details are provided in Appendix B.

n_k is drawn, consisting of IID input-label pairs $\{(\boldsymbol{x}_{ik}, y_{ik})\}_{i=1}^{n_k}$ where $y_{ik} \sim \mathcal{N}(\boldsymbol{\beta}_k^\top \boldsymbol{x}_{ik}, \sigma^2)$ to obtain sequence $\boldsymbol{Z}_k = \begin{bmatrix} \boldsymbol{x}_{1k} & \cdots & \boldsymbol{x}_{n_k,k} \\ y_{1k} & \cdots & y_{n_k,k} \end{bmatrix}^\top$. Let $n := \sum_{k=1}^K n_k$.

These sequences $(\boldsymbol{Z}_k)_{k=1}^K$, along with the query token $\boldsymbol{z}_{n+1} = [\boldsymbol{x}_{n+1} \ \ 0]^\top$, are concatenated to form a single prompt $\boldsymbol{Z}$. Recall (GLA), and let $f_{\mathsf{GLA}}(\boldsymbol{Z})$ denote the GLA prediction as defined in Theorem 1. Additionally, consider the model construction in (9), where $\boldsymbol{P}_q, \boldsymbol{P}_k \in \mathbb{R}^{d\times d}$ are trainable parameters. The GLA optimization problem is described as follows:

$$\begin{aligned} \mathcal{L}_{\mathsf{GLA}}^\star &:= \min_{\boldsymbol{P}_k, \boldsymbol{P}_q \in \mathbb{R}^{d\times d}, G\in\mathcal{G}} \mathcal{L}_{\mathsf{GLA}}(\boldsymbol{P}_k, \boldsymbol{P}_q, G), \\ \text{where} \quad \mathcal{L}_{\mathsf{GLA}}(\boldsymbol{P}_k, \boldsymbol{P}_q, G) &= \mathbb{E}\left[(y - f_{\mathsf{GLA}}(\boldsymbol{Z}))^2\right]. \end{aligned} \tag{18}$$

Here, $G(\cdot)$ represents the gating function and $\mathcal{G}$ denotes the function search space, which is determined by the chosen gating strategies (cf. Table 1 in Yang et al. (2023)).

Note that 1) the task vectors $(\boldsymbol{\beta}_k)_{k=1}^K$ are not explicitly shown in the prompt, 2) in-context features $\boldsymbol{x}_{ik}$ are randomly drawn, and 3) the gating function is applied to the tokens $(\boldsymbol{Z}_k)_{k=1}^K$. Given the above three evidences, the implicit weighting induced by the GLA model varies across different prompts, and it prevents the GLA from learning the optimal weighting. To address this, we introduce delimiters to mark the boundary of each task. Let $(\boldsymbol{d}_k)_{k=1}^K$ be the delimiters that determine stop of the tasks. Specifically, the final prompt is given by

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1^\top & \boldsymbol{d}_1 & \cdots & \boldsymbol{Z}_K^\top & \boldsymbol{d}_K & \boldsymbol{z}_{n+1} \end{bmatrix}^\top. \tag{19}$$

Additionally, to decouple the influence of gating and data, we envision that each token is $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top \ \ y_i \ \ \boldsymbol{c}_i^\top]^\top$ where $\boldsymbol{c}_i \neq \boldsymbol{0} \in \mathbb{R}^p$ is the contextual features with p being its dimension and $(\boldsymbol{x}_i, y_i)$ are the data features. Let $\bar{\boldsymbol{c}}_0, \cdots, \bar{\boldsymbol{c}}_K \in \mathbb{R}^p$ be $K+1$ contextual feature vectors.

- For task prompts $\boldsymbol{Z}_k$: Contextual features are set to a fixed vector $\bar{\boldsymbol{c}}_0 \neq \boldsymbol{0}$.
- For delimiters $\boldsymbol{d}_k$: Data features (e.g., $\boldsymbol{x}_i, y_i$) are set to zero so that $\boldsymbol{d}_k = [\boldsymbol{0}_{d+1}^\top \ \bar{\boldsymbol{c}}_k^\top]^\top$.

Explicit delimiters have been utilized in addressing real-world problems (Wang et al., 2024a; Asai et al., 2022; Dun et al., 2023) due to their ability to improve efficiency and enhance generalization, particularly in task-mixture or multi-document scenarios. To further motivate the introduction of $(\boldsymbol{d}_k)_{k=1}^K$, we present in Figure 1 the results of GLA training with and without delimiters, represented by the red and green curves, respectively. The black dashed curves indicate the optimal WPGD loss, $\mathcal{L}_{\mathsf{WPGD}}^\star$, under different scenarios. Notably, training GLA without delimiters (the green solid curve) performs strictly worse. In contrast, training with delimiters can achieve optimal performance under certain conditions (see Figures 1a, 1b, and 1c). Theorem 4, presented in the next section, provides a theoretical explanation for these observations, including the misalignment observed in Figure 1d.

5.2 Loss landscape of one-layer GLA

Given the input tokens with extended dimension, to ensure that GLA still implements WPGD as in Theorem 1, we propose the following model construction.

$$\tilde{\boldsymbol{W}}_k = \begin{bmatrix} \boldsymbol{W}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \tilde{\boldsymbol{W}}_q = \begin{bmatrix} \boldsymbol{W}_q & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \quad \text{and} \quad \tilde{\boldsymbol{W}}_v = \begin{bmatrix} \boldsymbol{W}_v & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{20}$$

Here, $\tilde{\boldsymbol{W}}_{k,q,v} \in \mathbb{R}^{(d+p+1)\times(d+p+1)}$ and $\boldsymbol{W}_{k,q,v} \in \mathbb{R}^{(d+1)\times(d+1)}$ are constructed via (9). The main idea is to set the last p rows and columns of attention matrices to zeros, ensuring that the delimiters do not affect the final prediction.

Assumption 2. *Contextual feature vectors $\bar{\boldsymbol{c}}_0, \cdots, \bar{\boldsymbol{c}}_K$ are linearly independent, and activation function $\phi(z) : \mathbb{R} \to [0, 1]$ is continuous, satisfying $\phi(-\infty) = 0$ and $\phi(+\infty) = 1$.*

Assumption 3. *The correlation between context tasks $(\boldsymbol{\beta}_k)_{k=1}^K$ and query task $\boldsymbol{\beta}$ satisfies $\mathbb{E}[\boldsymbol{\beta}_i^\top \boldsymbol{\beta}_j] = 0$ and $\mathbb{E}[\boldsymbol{\beta}_i^\top \boldsymbol{\beta}] \le \mathbb{E}[\boldsymbol{\beta}_j^\top \boldsymbol{\beta}]$ for all $1 \le i \le j \le K$.*

Given context examples $\{(\boldsymbol{X}_k, \boldsymbol{y}_k) := (\boldsymbol{x}_{ik}, y_{ik})_{i=1}^{n_k}\}_{k=1}^K$, define the concatenated data $(\boldsymbol{X}, \boldsymbol{y})$:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1^\top & \cdots & \boldsymbol{X}_K^\top \end{bmatrix}^\top \in \mathbb{R}^{n\times d}, \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1^\top & \cdots & \boldsymbol{y}_K^\top \end{bmatrix}^\top \in \mathbb{R}^n. \tag{21}$$

Based on the assumptions above, we are able to establish the equivalence between optimizing one-layer GLA and optimizing one-step WPGD predictor under scalar gating.

Theorem 4 (Scalar Gating)**.** *Suppose Assumption 2 holds. Recap the function $\mathcal{L}_{\mathsf{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega})$ from (2) with dataset $(\boldsymbol{X}, \boldsymbol{y})$ defined in (21). Consider (GLA) with input prompt $\boldsymbol{Z}$ defined in (19), the model constructions described in (20), and scalar gating $G(\boldsymbol{z}) = \phi(\boldsymbol{w}_g^\top \boldsymbol{z})\boldsymbol{1}_{(d+p+1)\times(d+p+1)}$, where $\boldsymbol{w}_g$ is a trainable parameter. Then, the optimal risk $\mathcal{L}^\star_{\mathsf{GLA}}$ defined in (18) obeys*

$$\mathcal{L}^\star_{\mathsf{GLA}} = \mathcal{L}^{\star,\mathcal{W}}_{\mathsf{WPGD}}, \quad \textit{where} \quad \mathcal{L}^{\star,\mathcal{W}}_{\mathsf{WPGD}} := \min_{\boldsymbol{P}\in\mathbb{R}^{d\times d}, \boldsymbol{\omega}\in\mathcal{W}} \mathcal{L}_{\mathsf{WPGD}}(\boldsymbol{P}, \boldsymbol{\omega}). \tag{22}$$

Here, $\mathcal{W} := \left\{ \begin{bmatrix} \omega_1 \boldsymbol{1}_{n_1}^\top & \cdots & \omega_K \boldsymbol{1}_{n_K}^\top \end{bmatrix}^\top \in \mathbb{R}^n \;\middle|\; 0 \le \omega_i \le \omega_j \le 1,\ \forall 1 \le i \le j \le K \right\}$.

Additionally, suppose Assumption 3 holds and $n_i = n_j$, for all $i, j \in [K]$. Let $\mathcal{L}^\star_{\mathsf{WPGD}}$ be the optimal WPGD risk (cf. (2)). Then $\mathcal{L}^\star_{\mathsf{GLA}}$ satisfies

$$\mathcal{L}^\star_{\mathsf{GLA}} = \mathcal{L}^\star_{\mathsf{WPGD}}. \tag{23}$$

Assumption 2 ensures that any $\boldsymbol{\omega}$ in $\mathcal{W}$ can be achieved by an appropriate choice of gating parameters. Furthermore, Assumption 3 guarantees that the optimal choice of $\boldsymbol{\omega}$ under the WPGD objective lies within the search space $\mathcal{W}$. The proof is provided in Appendix F.1.

In Figure 1, we conduct model training to validate our findings. Consider the setting where $K = 2$ and let $(r_1, r_2) = \left(\mathbb{E}[\boldsymbol{\beta}_1^\top \boldsymbol{\beta}]/d, \mathbb{E}[\boldsymbol{\beta}_2^\top \boldsymbol{\beta}]/d\right)$. In Figures 1a, 1b, and 1c, Assumption 3 holds, and the GLA results (shown in solid red) align with the optimal WPGD risk (represented by the dashed black), validating (23). However, in Figure 1d, since $r_1 > r_2$, Assumption 3 does not hold, and as a result, the optimal GLA loss $\mathcal{L}^\star_{\mathsf{GLA}}$ obtained from (22) is lower than the optimal WPGD loss $\mathcal{L}^\star_{\mathsf{WPGD}}$. Further details are deferred to Appendix B.

Loss landscape of vector gating. Till now, our discussion has focused on the scalar gating setting. It is important to highlight that, even in the scalar-weighting context, analyzing the WPGD problem remains non-trivial due to the joint optimization over $(\boldsymbol{P}, \boldsymbol{\omega})$. However, as demonstrated in Theorem 4, scalar gating can only express weightings within the set $\mathcal{W}$. If Assumption 3 does not hold, $\mathcal{L}^\star_{\mathsf{GLA}}$ cannot achieve the optimal WPGD loss (see the misalignment between red solid curve, presenting $\mathcal{L}^\star_{\mathsf{GLA}}$, and black dashed curve, presenting $\mathcal{L}^\star_{\mathsf{WPGD}}$ in Figure 1d). We argue that vector gating overcomes this limitation by applying distinct weighting mechanisms across different dimensions, facilitating stronger expressivity.

Theorem 5 (Vector Gating)**.** *Suppose Assumption 2 holds. Consider (GLA) with input prompt $\boldsymbol{Z}$ from (19), the model constructions from (20) but with $\tilde{\boldsymbol{W}}_v = \begin{bmatrix} \boldsymbol{0}_{(d+p+1)\times d} & \boldsymbol{u} & \boldsymbol{0}_{(d+p+1)\times p} \end{bmatrix}^\top$,*

and a vector gating $G(z) = \phi(W_g z)\mathbf{1}_{d+p+1}^\top$. *Recap Problem* (18) *with prediction defined as* $f_{\mathsf{GLA}}(Z) := o_{n+1}^\top h$, *where* $h \in \mathbb{R}^{d+p+1}$ *is a linear prediction head. Here,* u, W_g *and* h *are trainable parameters. Let* $\mathcal{L}^\star_{\mathsf{GLA}\text{-}v}$ *denote its optimal risk, and* $\mathcal{L}^\star_{\mathsf{WPGD}}$ *be defined as in* (2). *Then,* $\mathcal{L}^\star_{\mathsf{GLA}\text{-}v} = \mathcal{L}^\star_{\mathsf{WPGD}}$.

In Theorem 4, the equivalence between $\mathcal{L}^\star_{\mathsf{GLA}}$ and $\mathcal{L}^\star_{\mathsf{WPGD}}$ is established only when both Assumptions 2 and 3 are satisfied. In contrast, Theorem 5 demonstrates that applying vector gating requires only Assumption 2 to establish $\mathcal{L}^\star_{\mathsf{GLA}\text{-}v} = \mathcal{L}^\star_{\mathsf{WPGD}}$. Specifically, under the bounded activation model of Assumption 2, scalar gating is unable to express non-monotonic weighting schemes. For instance, suppose there are two tasks: Even if Task 1 is more relevant to the query, Assumption 2 will assign a higher (or equal) weight to examples in Task 2 resulting in sub-optimal prediction. Theorem 5 shows that vector gating can avoid such bottlenecks by potentially encoding tasks in distinct subspaces. To verify these intuitions, in Figure 1d, we train a GLA model with vector gating and results are presented in cyan curve, which outperform the scalar gating results (red solid) and align with the optimal WPGD loss (black dashed).

Loss landscape of one-layer linear attention. Given the fact that linear attention is equivalent to (GLA) when implemented with all ones gating, that is, $G_i \equiv \mathbf{1}$, we derive the following corollary. Consider training a standard single-layer linear attention, i.e., by setting $G_i \equiv \mathbf{1}$ in (GLA), and let $f_{\mathsf{ATT}}(Z)$ be its prediction. Let $\mathcal{L}^\star_{\mathsf{ATT}}$ be the corresponding optimal risk following (18).

Corollary 2. *Consider a single-layer linear attention following model construction in* (9) *and let linear data as given in Definition 2. Let* R *and* r *be the corresponding correlation matrix and vector as defined in Definition 1. Suppose* $\Sigma = I$. *Then, the optimal risk obeys*

$$\mathcal{L}^\star_{\mathsf{ATT}} := \min_{P \in \mathbb{R}^{d\times d}} \mathcal{L}_{\mathsf{WPGD}}(P, \omega = \mathbf{1}) = d + \sigma^2 - \frac{d(\mathbf{1}^\top r)^2}{n(d+\sigma^2+1) + \mathbf{1}^\top R \mathbf{1}}. \tag{24}$$

In the Figure 1, blue solid curves represent the linear attention results and blue dashed are the theory curves following (24). The two curves are aligned in all the subfigures, which validate our Corollary 2. More implementation details are deferred to Appendix B.

6 Discussion

Our analysis is currently limited to GLA models under specific weight construction assumptions as outlined in Section 3, which may not capture the full expressivity of practical GLA implementations. The theoretical framework we developed requires certain structural constraints on the attention matrices (as specified in Equation 9) to establish the connection with WPGD algorithms.

Several important directions for future research include:

1. Extending our analysis to more general GLA architectures without the weight construction constraints, which would better reflect deployed models such as Mamba (Gu & Dao, 2023; Dao & Gu, 2024) and RWKV (Peng et al., 2024).
2. Investigating when delimiters are necessary for effective learning in multi-task settings. Our experiments show that without delimiters, GLA models perform sub-optimally, but a deeper theoretical characterization of this phenomenon is needed.
3. Exploring the GLA landscape where gating functions depend on input features in more complex ways than our current formulation allows.
4. Analyzing the effects of incorporating MLP layers between GLA layers, which could enhance the model's expressive power beyond what we have characterized.

Addressing these limitations would further bridge the gap between our theoretical understanding and the practical performance of state-of-the-art GLA-based sequence models.

Acknowledgements

Y.L. and S.O. were supported in part by the NSF grants CCF2046816, CCF-2403075, CCF-2008020, the Office of Naval Research grant N000142412289, and by gifts/awards from Open Philanthropy, OpenAI, Amazon Research, and Google Research. M.F. was supported in part by awards NSF TRIPODS II 2023166, CCF 2007036, CCF 2212261, and CCF 2312775.

References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*, 2024.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. *arXiv preprint arXiv:2205.11961*, 2022.

Han Bao, Ryuichiro Hataya, and Ryo Karakida. Self-attention networks localize when qk-eigenspectrum concentrates. *arXiv preprint arXiv:2402.02098*, 2024.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Atlas: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025a.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. *arXiv preprint arXiv:2504.13173*, 2025b.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Sitan Chen and Yuanzhi Li. Provably learning a multi-head attention layer. *arXiv preprint arXiv:2402.04084*, 2024.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

Liam Collins, Advait Parulekar, Aryan Mokhtari, Sujay Sanghavi, and Sanjay Shakkottai. In-context learning with transformers: Softmax attention adapts to function lipschitzness. *arXiv preprint arXiv:2402.11639*, 2024.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

Yichuan Deng, Zhao Song, Shenghao Xie, and Chiwun Yang. Unmasking transformers: A theoretical approach to data recovery via attention weights. *arXiv preprint arXiv:2310.12462*, 2023.

Puneesh Deora, Rouzbeh Ghaderi, Hossein Taheri, and Christos Thrampoulidis. On the optimization and generalization of multi-head attention. *arXiv preprint arXiv:2310.12680*, 2023.

Nan Ding, Tomer Levinboim, Jialin Wu, Sebastian Goodman, and Radu Soricut. Causallm is not optimal for in-context learning. *arXiv preprint arXiv:2308.06912*, 2023.

Chen Dun, Mirian Hipolito Garcia, Guoqing Zheng, Ahmed Hassan Awadallah, Anastasios Kyrillidis, and Robert Sim. Sweeping heterogeneity with smart mops: Mixture of prompts for llm task adaptation. *arXiv preprint arXiv:2310.02842*, 2023.

Tolga Ergen, Behnam Neyshabur, and Harsh Mehta. Convexifying transformers: Improving optimization and understanding of transformer networks. *arXiv:2211.11052*, 2022.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Deqing Fu, Tianqi Chen, Robin Jia, and Vatsal Sharan. Transformers learn higher-order optimization methods for in-context learning: A study with linear models. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Khashayar Gatmiry, Nikunj Saunshi, Sashank Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? In *Proceedings of the 41st International Conference on Machine Learning*, pp. 15130–15152, 2024.

Riccardo Grazzi, Julien Siems, Simon Schrodi, Thomas Brox, and Frank Hutter. Is mamba capable of in-context learning? *arXiv preprint arXiv:2402.03170*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Ruiquan Huang, Yingbin Liang, and Jing Yang. Non-asymptotic convergence of training transformers for next-token prediction. *arXiv preprint arXiv:2409.17335*, 2024.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023.

M Emrullah Ildiz, Yixiao Huang, Yingcong Li, Ankit Singh Rawat, and Samet Oymak. From self-attention to markov models: Unveiling the dynamics of generative transformers. *arXiv preprint arXiv:2402.13512*, 2024.

Samy Jelassi, Michael Eli Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.

Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis of in-context learning. *arXiv preprint arXiv:2401.15530*, 2024.

Aaron Alvarado Kristanto Julistiono, Davoud Ataee Tarzanagh, and Navid Azizan. Optimizing attention with mirror descent: Generalized max-margin token selection. *arXiv preprint arXiv:2410.14581*, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Tobias Katsch. Gateloop: Fully data-controlled linear recurrence for sequence modeling. *arXiv preprint arXiv:2311.01927*, 2023.

Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. *arXiv preprint arXiv:2302.06015*, 2023a.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pp. 19565–19594. PMLR, 2023b.

Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 685–693. PMLR, 2024a.

Yingcong Li, Ankit Singh Rawat, and Samet Oymak. Fine-grained analysis of in-context linear estimation: Data, architecture, and beyond. *arXiv preprint arXiv:2407.10005*, 2024b.

Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning. *arXiv preprint arXiv:2402.18819*, 2024.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains. *arXiv preprint arXiv:2402.04161*, 2024.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023.

Samet Oymak, Ankit Singh Rawat, Mahdi Soltanolkotabi, and Christos Thrampoulidis. On the role of attention in prompt-tuning. In *International Conference on Machine Learning*, 2023.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can mamba learn how to learn? a comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*, 2024.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Teddy Ferdinan, Haowen Hou, Przemysław Kazienko, et al. Eagle and finch: RWKV with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*, 2024.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*, 2024.

Arda Sahiner, Tolga Ergen, Batu Ozturkler, John Pauly, Morteza Mardani, and Mert Pilanci. Unraveling attention via convex duality: Analysis and interpretations of vision transformers. In *International Conference on Machine Learning*, pp. 19050–19088. PMLR, 2022.

Heejune Sheen, Siyu Chen, Tianhao Wang, and Harrison H Zhou. Implicit regularization of gradient flow on one-layer softmax attention. *arXiv preprint arXiv:2403.08699*, 2024.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 44712–44740, 2024.

Jerome Sieber, Carmen Amo Alonso, Alexandre Didier, Melanie N Zeilinger, and Antonio Orvieto. Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks. *arXiv preprint arXiv:2405.15731*, 2024.

Haoyuan Sun, Ali Jadbabaie, and Navid Azizan. In-context learning of polynomial kernel regression in transformers with glu layers. *arXiv preprint arXiv:2501.18187*, 2025.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models. *arXiv preprint arXiv:2405.05254*, 2024.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism. *Advances in Neural Information Processing Systems*, 36, 2024.

Christos Thrampoulidis. Implicit bias of next-token prediction. *arXiv preprint arXiv:2402.18551*, 2024.

Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *arXiv preprint arXiv:2310.00535*, 2023.

Bhavya Vasudeva, Puneesh Deora, and Christos Thrampoulidis. Implicit bias and fast convergence rates for self-attention. *arXiv preprint arXiv:2402.05738*, 2024a.

Bhavya Vasudeva, Deqing Fu, Tianyi Zhou, Elliott Kau, Youqi Huang, and Vatsal Sharan. Simplicity bias of transformers to learn low sensitivity functions. *arXiv preprint arXiv:2403.06925*, 2024b.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Johannes von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Max Vladymyrov, Razvan Pascanu, et al. Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*, 2023.

Ruochen Wang, Sohyun An, Minhao Cheng, Tianyi Zhou, Sung Ju Hwang, and Cho-Jui Hsieh. One prompt is not enough: Automated construction of a mixture-of-expert prompts. *arXiv preprint arXiv:2407.00256*, 2024a.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. *arXiv preprint arXiv:2406.06893*, 2024b.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522, 2024b.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Yize Zhao and Christos Thrampoulidis. Geometry of concepts in next-token prediction: Neural-collapse meets semantics. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025. URL https://openreview.net/forum?id=Lt6l1woQ84.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. *arXiv preprint arXiv:2408.15417*, 2024.

Junhao Zheng, Shengjie Qiu, and Qianli Ma. Learn or recall? revisiting incremental learning with pre-trained language models. *arXiv preprint arXiv:2312.07887*, 2023.

Itamar Zimerman, Ameen Ali, and Lior Wolf. A unified implicit attention formulation for gated-linear recurrent sequence models. *arXiv preprint arXiv:2405.16504*, 2024.

Chang Zong, Jian Shao, Weiming Lu, and Yueting Zhuang. Stock movement prediction with multimodal stable fusion via gated cross-attention mechanism. *arXiv preprint arXiv:2406.06594*, 2024.

A Capabilities of multi-layer GLA

Ahn et al. (2024) demonstrated that, with appropriate construction, an L-layer linear attention model performs L-step preconditioned GD on the dataset $(\boldsymbol{x}_i, y_i)_{i=1}^n$ provided within the prompt. In this work, we study multi-layer GLA and analyze the associated algorithm class it can emulate. It is worth mentioning that Ahn et al. (2024) does not consider *causal masking* which is integral to multilayer GLA due to its recurrent nature described in (GLA). Our analysis will capture the impact of gating and causal mask through n separate GD trajectories that are coupled.

Multi-layer GLA. For any input prompt $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times(d+1)}$, let $\texttt{GLA}(\boldsymbol{Z}) := [\boldsymbol{o}_1 \ \boldsymbol{o}_2 \ \cdots \ \boldsymbol{o}_{n+1}]^\top \in \mathbb{R}^{(n+1)\times(d+1)}$ denote its GLA output, as defined in (GLA). We consider an L-layer GLA model as follows: For each layer $\ell \in [L]$, let $\boldsymbol{Z}_\ell$ denote its input, where we set $\boldsymbol{Z}_1 = \boldsymbol{Z}$, and let $\boldsymbol{W}_{k,\ell}, \boldsymbol{W}_{q,\ell}, \boldsymbol{W}_{v,\ell}$ denote the key, query, and value matrices for layer ℓ, respectively. After applying a residual connection, the input of the ℓ-th layer is given by

$$\boldsymbol{Z}_\ell := \boldsymbol{Z}_{\ell-1} + \texttt{GLA}_{\ell-1}(\boldsymbol{Z}_{\ell-1}). \tag{25}$$

Here, $\texttt{GLA}_\ell(\cdot)$ denotes the output of the ℓ-th GLA layer, associated with the attention matrices $(\boldsymbol{W}_{q,\ell}, \boldsymbol{W}_{k,\ell}, \boldsymbol{W}_{v,\ell})$ for all $\ell \in [L]$. In the following, we focus on the $(d+1)$-th entry of each token's output at each layer (after the residual connection), denoted by $o_{i,\ell} := [\boldsymbol{Z}_\ell + \texttt{GLA}_\ell(\boldsymbol{Z}_\ell)]_{i,d+1}$ for all $i \in [n+1]$ and $\ell \in [L]$.

Theorem 6. *Consider an L-layer GLA defined in (25), where $\boldsymbol{W}_{k,\ell}, \boldsymbol{W}_{q,\ell}$ in the ℓ'th layer parameterized by $\boldsymbol{P}_{k,\ell}, -\boldsymbol{P}_{q,\ell} \in \mathbb{R}^{d\times d}$, for all $\ell \in [L]$ following (9). Let the gating be a function of the features, e.g., $\boldsymbol{G}_i = G(\boldsymbol{x}_i)$, and define $\boldsymbol{\Omega}$ as in Theorem 1. Additionally, denote the masking as $\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{I}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{n\times n}$, and let $\hat{\boldsymbol{\beta}}_0 = \boldsymbol{\beta}_{i,0} = \boldsymbol{0}$ for all $i \in [n]$. Recall that $o_{i,\ell}$ for all $i \in [n+1]$ and $\ell \in [L]$ stands for the last entry of the i'th token output at the ℓ'th layer. We have*

- *For all $i \le n$, $o_{i,\ell} = y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_{i,\ell}$, where $\boldsymbol{\beta}_{i,\ell} = \boldsymbol{\beta}_{i,\ell-1} - \boldsymbol{P}_{q,\ell}\left(\nabla_{i,\ell} \oslash \boldsymbol{g}_{i:n+1}\right)$;*
- *$o_{n+1,\ell} = -\boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\ell$, where $\hat{\boldsymbol{\beta}}_\ell = (1-\alpha_\ell)\hat{\boldsymbol{\beta}}_{\ell-1} - \boldsymbol{P}_{q,\ell}\left(\nabla_{n,\ell} \oslash \boldsymbol{g}_{n+1}\right)$, and $\alpha_\ell = \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell}\boldsymbol{P}_{k,\ell}^\top \boldsymbol{x}$.*

Here, letting $\boldsymbol{B}_\ell = [\boldsymbol{\beta}_{1,\ell} \ \cdots \ \boldsymbol{\beta}_{n,\ell}]^\top$, $\boldsymbol{X}_\ell = \boldsymbol{X}\boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}$, and $\boldsymbol{y}_\ell = \mathrm{diag}(\boldsymbol{X}\boldsymbol{B}_\ell^\top)$, we define

$$\nabla_{i,\ell} = \boldsymbol{X}_\ell^\top \boldsymbol{M}_i \left(\boldsymbol{y}_\ell - \boldsymbol{y}\right).$$

We defer the proof of Theorem 6 to the Appendix D.2. Theorem 6 states that L-layer GLA (25) implements L steps of WPGD with gradient in a recurrent form and additional weight decay α_ℓ. To recap, given data $(\boldsymbol{X}, \boldsymbol{y})$ and prediction $\hat{\boldsymbol{\beta}}$, the gradient with respect to the squared loss takes the form $\boldsymbol{X}^\top(\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{y})$, up to some constant c. In comparison, $\boldsymbol{P}_{q,\ell}\left(\nabla_{i,\ell} \oslash \boldsymbol{g}_{i:n+1}\right)$ similarly acts as a gradient but incorporates layer-wise feature preconditioners $(\boldsymbol{P}_{q,\ell}, \boldsymbol{P}_{k,\ell})$, data weighting $(\boldsymbol{\Omega})$, and causality $(\boldsymbol{g}_{i:n+1}, \boldsymbol{M}_i)$. Here, $\boldsymbol{M}_i$ represents causal masking, ensuring that at time i, only inputs from $j \le i$ are used for prediction. Notably, the recurrent structure of GLA allows the gating mechanism to apply context-dependent weighting strategies.

To simplify the theorem statement, we assume that the gating function depends only on the input feature, e.g., $\boldsymbol{G}_i = G(\boldsymbol{x}_i)$, ensuring that the corresponding data-dependent weighting is uniform across all layers. This assumption is included solely for clarity in the theorem statement, and if the gating function varies across layers or depends on the entire $(d+1)$-dimensional token rather than just the feature $\boldsymbol{x}_i$, each layer has its own gating $\boldsymbol{\Omega}_\ell$ and $\boldsymbol{X}_\ell = \boldsymbol{X}\boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}_\ell$. Note that our inclusion of the additional term α_ℓ captures the influence of the last token's output on the next layer's prediction. Based on the above L-layer GLA result, we have the following corollary for multi-layer linear attention network with causal mask in each layer.

Corollary 3. *Consider an L-layer linear attention model with causal mask and residual connection in each layer. Let ℓ'th layer follows the weight construction in (9) with $\boldsymbol{W}_{k,\ell}, \boldsymbol{W}_{q,\ell}$ parameterized by $\boldsymbol{P}_{k,\ell}, -\boldsymbol{P}_{q,\ell}$ as in Theorem 6 and define $\boldsymbol{P}_\ell := \boldsymbol{P}_{q,\ell}\boldsymbol{P}_{k,\ell}^\top$, for $\ell \in [L]$. Let $\hat{\boldsymbol{\beta}}_0 = \boldsymbol{0}$. Then, the $(d+1)$'th entry of the last token output of each layer satisfies:*

$$o_{n+1,\ell} = -\boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\ell, \quad \textit{where} \quad \hat{\boldsymbol{\beta}}_\ell = (1-\alpha_\ell)\hat{\boldsymbol{\beta}}_{\ell-1} - \boldsymbol{P}_\ell \boldsymbol{X}^\top(\boldsymbol{y}_\ell - \boldsymbol{y}).$$

Here, we define $\alpha_\ell = \boldsymbol{x}^\top \boldsymbol{P}_\ell \boldsymbol{x}$ and $\boldsymbol{y}_\ell$ follows the same definition as in Theorem 6.

Note that Corollary 3 generalizes (Ahn et al., 2024, Lemma 1) by extending it to causal attention. When considering full attention (i.e., without applying the causal mask), we have $\boldsymbol{y}_\ell = \boldsymbol{X}\hat{\boldsymbol{\beta}}_\ell$, and the expression $\boldsymbol{X}^\top(\boldsymbol{y}_\ell - \boldsymbol{y}) = \boldsymbol{X}^\top(\boldsymbol{X}\hat{\boldsymbol{\beta}}_\ell - \boldsymbol{y})$ corresponds to the standard gradient of the squared loss. Furthermore, if we disregard the impact of the last token (e.g., by incorporating the matrix M as in Eq. (3) of Ahn et al. (2024)), then $\alpha_\ell = 0$. These results are also consistent with Ding et al. (2023), which demonstrate that causal masking limits convergence by introducing sequence biases, akin to online gradient descent with non-decaying step sizes.

Our theoretical results in Theorem 6 focus on multi-layer GLA without Multi-Layer Perceptron (MLP) layers to isolate and analyze the effects of the gating mechanism. However, MLP layers, a key component of standard Transformers, facilitate further nonlinear feature transformations and interactions, potentially enhancing GLA's expressive power. Future work could explore the theoretical foundations of integrating MLPs into GLA and analyze the optimization landscape of general gated attention models, aligning them more closely with conventional Transformer architectures (Gu & Dao, 2023; Dao & Gu, 2024; Peng et al., 2024).

B Experimental setup

B.1 Implementation detail

Data generation. Consider ICL problem with input in the form of multi-task prompt as described in Section 5.1. In the experiments, we set $K = 2$, dimensions $d = 10$ and $p = 5$, uniform context length $n_1 = n_2 = \bar{n}$ where we have $n = K\bar{n}$, and vary $\bar{n}$ from 0 to 50. Let $(r_1, r_2) := \left(\mathbb{E}[\boldsymbol{\beta}_1^\top \boldsymbol{\beta}]/d, \mathbb{E}[\boldsymbol{\beta}_2^\top \boldsymbol{\beta}]/d\right)$ denote the correlations between in-context tasks $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ and query task $\boldsymbol{\beta}$. We generate task vectors as follows:

$$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \sim \mathcal{N}(0, \boldsymbol{I}_d), \quad \text{and} \quad \boldsymbol{\beta} \sim \mathcal{N}(r_1\boldsymbol{\beta}_1 + r_2\boldsymbol{\beta}_2, (1 - r_1^2 - r_2^2)\boldsymbol{I}_d).$$

Input features are randomly sampled $((\boldsymbol{x}_{ik})_{i=1}^{\bar{n}})_{k=1}^{K}, \boldsymbol{x}_{n+1} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, and we have $y_{ik} = \boldsymbol{\beta}_k^\top \boldsymbol{x}_{ik}$ ($\sigma = 0$), $k \in \{1, 2\}$ and $y_{n+1} = \boldsymbol{\beta}^\top \boldsymbol{x}_{n+1}$. Additionally, delimiters $\bar{\boldsymbol{c}}_0, \cdots, \bar{\boldsymbol{c}}_K$ are randomly sampled from $\mathcal{N}(0, \boldsymbol{I}_p)$.

Implementation setting. We train 1-layer linear attention and GLA models for solving multi-prompt ICL problem as described in Section 5.1. For GLA model, we consider sigmoid-type gating function given by scalar gating: $G(\boldsymbol{z}) = \phi(\boldsymbol{w}_g^\top \boldsymbol{z})\boldsymbol{1}_{(d+p+1)\times(d+p+1)}$, or vector gating: $G(\boldsymbol{z}) = \phi(\boldsymbol{W}_g \boldsymbol{z})\boldsymbol{1}_{d+p+1}^\top$ where $\phi(z) = (1 + e^{-z})^{-1}$ is the activation function. Note that although the theoretical results are based on the model constructions (c.f. (9) and (20)), we do not restrict the attention weights in our implementation. We train each model for 10000 iterations with batch size 256 and Adam optimizer with learning rate 10^{-3}. Similar to the previous work (Li et al., 2024b), since our study focuses on the optimization landscape, ICL problems using linear attention/GLA models are non-convex, and experiments are implemented via gradient descent, we repeat 10 model trainings from different model initialization and data sampling (e.g., different choice of delimiters) and results are presented as the minimal test risk among those 10 trails. Results presented have been normalized by d.

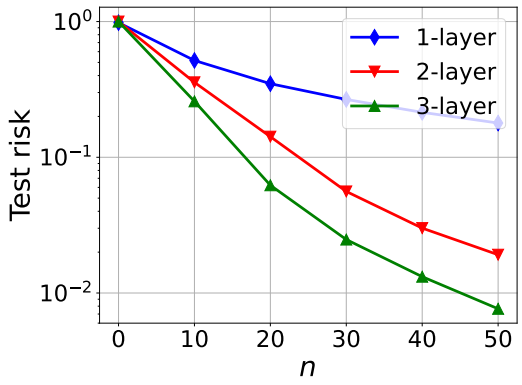


Figure 2: Multi-layer GLA experiments with $(r_1, r_2) = (0, 1)$.

Experimental results. Based on the experimental setting, we can obtain the correlation matrix and vector following Definition 1

$$\boldsymbol{R} = \begin{bmatrix} \mathbf{1}_{\tilde{n}}\mathbf{1}_{\tilde{n}}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{\tilde{n}}\mathbf{1}_{\tilde{n}}^\top \end{bmatrix} \quad \text{and} \quad \boldsymbol{r} = \begin{bmatrix} r_1\mathbf{1}_{\tilde{n}}^\top & r_2\mathbf{1}_{\tilde{n}}^\top \end{bmatrix}^\top.$$

Then dotted curves display our theoretical results derive using $\boldsymbol{\Sigma} = \boldsymbol{I}$ and $\boldsymbol{R}, \boldsymbol{r}$ above. Specifically, in Figure 1, black dashed curves represent $\mathcal{L}^\star_{\texttt{WPGD}}$ following (17) and blues dashed curves represent $\mathcal{L}^\star_{\texttt{GLA}}$ following (24). We consider scenarios where $(r_1, r_2) \in \{(0,1), (0.2, 0.8), (0.5, 0.5), (0.8, 0.2)\}$ and results are presented in Figures (1a), (1b), (1c) and (1d), respectively.

- `GLA-wo` achieves the worst performance among all the methods. We claim that it is due to the randomness of input tokens as discussed in Section 5.1. Thanks to the introduction of delimiters as described in (19), data and gating is decoupled and a task-dependent weighting is learnt. Hence, `GLA` is able to achieve comparable performance to the optimal one ($\mathcal{L}^\star_{\texttt{WPGD}}$, red dashed). Note that `GLA-wo` performs even worse than `LinAtt`. It comes from the fact the weighting induced by `GLA-wo` varies over different input prompts and it can not implement all ones weight.

- The alignments between `LinAtt` (blue solid) and blue dashed curves validate our Corollary 2. In Figures 1a, 1b and 1c, the alignments between `GLA` (red solid) and $\mathcal{L}_{\texttt{WPGD}}$ (black dashed) verify our Theorem 4, specifically, Equation 23. While in 1c and 1d, `GLA` achieves the same performance as `LinAtt`. It is due to the fact that `GLA` can not weight the history higher than its present. Then the equal-weighting, e.g., $\boldsymbol{\omega} = \mathbf{1}$, is the optimal weighting given such constraint. What's more, the alignment between `GLA-vector` (cyan curves) and red dashed in Figure 1d validates our vector gating theorem in Theorem 5.

B.2 Multi-layer experiments

In this section, we present additional experiments on multi-layer GLA models. We adopt the same experimental setup as described in Figure 1a and Appendix B, with parameters setting to $(r_1, r_2) = (0, 1)$. The results are displayed in Figure 2, where the blue, red, and green curves correspond to the performance of one-, two-, and three-layer GLA models, respectively, with the y-axis presented in log-scale. According to Theorem 6, an L-layer GLA performs L steps of WPGD, suggesting that deeper models should yield improved predictive performance. The experimental findings in Figure 2 align with the theoretical predictions of Theorem 6.

C Related work

Efficient sequence models. Recent sequence model proposals – such as RetNet (Sun et al., 2023), Mamba (Gu & Dao, 2023), xLSTM (Beck et al., 2024), GLA Transformer (Yang et al.,

2023), RWKV-6 (Peng et al., 2024) – admit efficient recurrent forms while being increasingly competitive with the transformer architecture with softmax-attention. However, we have a rather limited theoretical understanding of these architectures, especially, when it comes to their optimization landscape and ICL capabilities. Park et al. (2024); Grazzi et al. (2024) demonstrate that Mamba is effective and competitive with a transformer of similar size in various ICL tasks, whereas Arora et al. (2024); Jelassi et al. (2024) establish theoretical and empirical shortcomings of recurrent models for solving recall tasks. It is worth mentioning that, GLA models also connect to state-space models and linear RNNs (De et al., 2024; Orvieto et al., 2023; Gu et al., 2021; Fu et al., 2022), as they could be viewed as time-varying SSMs (Dao & Gu, 2024; Sieber et al., 2024). Finally, GLA models are also closely related to implicit self-attention frameworks. For example, the work by Zimerman et al. (2024) on unified implicit attention highlights how models such as Mamba (Gu & Dao, 2023) and RWKV (Peng et al., 2023) can be viewed under a shared attention mechanism. Additionally, Zong et al. (2024) leverage gated cross-attention for robust multimodal fusion, demonstrating another practical application of gated mechanisms. Both approaches align with GLA's data-dependent gating, suggesting its potential for explainability and stable fusion tasks.

Theory of in-context learning. The theoretical aspects of ICL has been studied by a growing body of works during the past few years (Xie et al., 2022; von Oswald et al., 2023; Gatmiry et al., 2024; Li et al., 2023b; Collins et al., 2024; Wu et al., 2023; Fu et al., 2023; Lin & Lee, 2024; Akyürek et al., 2023; Zhang et al., 2024). A subset of these follow the setting of Garg et al. (2022) which investigates the ICL ability of transformers by focusing on prompts where each example is labeled by a task function from a specific function class, such as linear models. Akyürek et al. (2023) focuses on linear regression and provide a transformer construction that can perform a single step of GD based on in-context examples. Similarly, Von Oswald et al. (2023) provide a construction of weights in linear attention-only transformers that can replicate GD steps for a linear regression task on in-context examples. Notably, they observe similarities between their constructed networks and those resulting from training on ICL prompts for linear regression tasks. Building on these, Zhang et al. (2024); Mahankali et al. (2023); Ahn et al. (2024) focus on the loss landscape of ICL for linear attention models. For a single-layer model trained on in-context prompts for random linear regression tasks, Mahankali et al. (2023); Ahn et al. (2024) show that the resulting model performs a single preconditioned GD step on in-context examples in a test prompt, aligning with the findings of Von Oswald et al. (2023). In contrast, real-world LLMs incorporate additional components (e.g., softmax attention, causal masking, MLPs) that enable more sophisticated learning and lead to expected performance gaps (Shen et al., 2024). GLA models helped bridge the gap from linear attention to transformers. More recent work (Ding et al., 2023) analyzes the challenges of causal masking in causal language models (causalLM), showing that their suboptimal convergence dynamics closely resemble those of online gradient descent with non-decaying step sizes. Additionally, Li et al. (2024b) analyzes the landscape of the H3 architecture, an SSM, under the same dataset model. They show that H3 can implement WPGD thanks to its convolutional/SSM filter. However, their WPGD theory is restricted to the trivial setting of equal weights, relying on the standard prompt model with IID examples and shared tasks. In contrast, we propose novel multitask datasets and prompt models where nontrivial weighting is provably optimal. This allows us to characterize the loss landscape of WPGD and explore advanced GLA models, linking them to data-dependent WPGD algorithms.

Optimization landscape of attention mechanisms. The optimization behavior of attention mechanisms has become a rapidly growing area of research (Deora et al., 2023; Huang et al., 2023; Tian et al., 2023; Fu et al., 2023; Li et al., 2024a; Tarzanagh et al., 2024; 2023; Deng et al., 2023; Makkuva et al., 2024; Jeon et al., 2024; Zheng et al., 2023; Collins et al., 2024; Chen & Li, 2024; Li et al., 2023a; Ildiz et al., 2024; Vasudeva et al., 2024b; Bao et al., 2024; Chen et al., 2024; Huang et al., 2024; Wang et al., 2024b; Sun et al., 2025). Particularly relevant to our work are studies investigating convex relaxation approaches to optimize attention models (Sahiner et al., 2022; Ergen et al., 2022), gradient-based optimization methods in vision transformers (Jelassi et al., 2022), optimization dynamics in prompt-attention (Oymak et al., 2023; Tarzanagh et al., 2024), implicit bias of gradient descent

for attention optimization (Tarzanagh et al., 2024; 2023; Julistiono et al., 2024; Li et al., 2024a; Thrampoulidis, 2024; Zhao et al., 2024; Vasudeva et al., 2024a; Sheen et al., 2024), and optimization geometry analysis of next-token prediction (Zhao & Thrampoulidis, 2025). While these works mainly focus on standard attention, our work provides the first optimization landscape analysis of *gated* attention, establishing conditions for a unique global minimum and providing comprehensive insights into the implications of data-dependent weighting in the optimization dynamics encountered during training.

D GLA $\Leftrightarrow$ WPGD

D.1 Proof of Theroem 1

Proof. Recap the problem settings from Section 2 where in-context samples are given by

$$\boldsymbol{Z} = [z_1 \ \cdots \ z_n \ z_{n+1}]^\top = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n & \boldsymbol{x}_{n+1} \\ y_1 & \cdots & y_n & 0 \end{bmatrix}^\top$$

and let the value, key and query embeddings at time i be

$$\boldsymbol{v}_i = \boldsymbol{W}_v{}^\top \boldsymbol{z}_i, \quad \boldsymbol{k}_i = \boldsymbol{W}_k{}^\top \boldsymbol{z}_i, \quad \text{and} \quad \boldsymbol{q}_i = \boldsymbol{W}_q{}^\top \boldsymbol{z}_i.$$

Then we can rewrite the GLA output (cf. (GLA)) as follows:

$$\begin{aligned} \boldsymbol{o}_i = \boldsymbol{S}_i \boldsymbol{q}_i \quad \text{and} \quad \boldsymbol{S}_i &= \boldsymbol{G}_i \odot \boldsymbol{S}_{i-1} + \boldsymbol{v}_i \boldsymbol{k}_i^\top \\ &= \boldsymbol{G}_i \odot \cdots \boldsymbol{G}_1 \odot \boldsymbol{v}_1 \boldsymbol{k}_1^\top + \cdots + \boldsymbol{G}_i \odot \boldsymbol{v}_{i-1} \boldsymbol{k}_{i-1}^\top + \boldsymbol{v}_i \boldsymbol{k}_i^\top \\ &= \sum_{j=1}^{i} \boldsymbol{G}_{j:i} \odot \boldsymbol{v}_j \boldsymbol{k}_j^\top, \end{aligned}$$

where we define

$$\boldsymbol{G}_{j:i} = \boldsymbol{G}_{j+1} \odot \boldsymbol{G}_{j+2} \cdots \boldsymbol{G}_i, \quad j < i, \quad \text{and} \quad \boldsymbol{G}_{i:i} = \boldsymbol{1}_{(d+1)\times(d+1)}.$$

Consider the prediction based on the last token, then we obtain

$$\boldsymbol{o}_{n+1} = \boldsymbol{S}_{n+1} \boldsymbol{q}_{n+1} \quad \text{and} \quad \boldsymbol{S}_{n+1} = \sum_{j=1}^{n+1} \boldsymbol{G}_{j:n+1} \odot \boldsymbol{v}_j \boldsymbol{k}_j^\top.$$

Construction 1: Recall the model construction from (9) where

$$\boldsymbol{W}_k = \begin{bmatrix} \boldsymbol{P}_k & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix}, \quad \boldsymbol{W}_q = \begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{W}_v = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{26}$$

Then, given each token $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top \ y_i]^\top$, $i \in [n]$, single-layer GLA returns

$$\boldsymbol{v}_i = \begin{bmatrix} \boldsymbol{0} \\ y_i \end{bmatrix}, \quad \boldsymbol{k}_i = \begin{bmatrix} \boldsymbol{P}_k^\top \boldsymbol{x}_i \\ 0 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{q}_i = \begin{bmatrix} \boldsymbol{P}_q^\top \boldsymbol{x}_i \\ 0 \end{bmatrix},$$

and we obtain

$$\boldsymbol{v}_i \boldsymbol{k}_i^\top = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ y_i \boldsymbol{x}_i^\top \boldsymbol{P}_k & 0 \end{bmatrix}, \quad i \le n, \quad \text{and} \quad \boldsymbol{v}_{n+1} \boldsymbol{k}_{n+1}^\top = \boldsymbol{0}_{(d+1)\times(d+1)}.$$

Therefore, since only d entries in $\boldsymbol{v}_i \boldsymbol{k}_i^\top$ matrix are nonzero, given $\odot$ as the Hadamard product, only the corresponding d entries in all $\boldsymbol{G}_i$ matrices are useful. Based on this observation, let

$$\boldsymbol{G}_i = \begin{bmatrix} * & * \\ \boldsymbol{g}_i^\top & * \end{bmatrix} \quad \text{and} \quad \boldsymbol{G}_{j:i} = \begin{bmatrix} * & * \\ \boldsymbol{g}_{j:i}^\top & *, \end{bmatrix}$$

where $\boldsymbol{g}_{j:i} = \boldsymbol{g}_{j+1} \odot \boldsymbol{g}_{j+2} \cdots \boldsymbol{g}_i \in \mathbb{R}^d$ for $j < i$ and $\boldsymbol{g}_{i:i} = \boldsymbol{1}_d$.

Combing all together, and letting $\boldsymbol{X}, \boldsymbol{y}$ follow the definitions in (6), we obtain

$$\begin{aligned}
\boldsymbol{o}_{n+1} &= \boldsymbol{S}_{n+1}\boldsymbol{q}_{n+1} \\
&= \left(\sum_{j=1}^{n+1} \boldsymbol{G}_{j:n+1} \odot \boldsymbol{v}_j \boldsymbol{k}_j^\top\right) \boldsymbol{q}_{n+1} \\
&= \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \sum_{j=1}^{n} y_j \boldsymbol{x}_j^\top \boldsymbol{P}_k \odot \boldsymbol{g}_{j:n+1}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_q^\top \boldsymbol{x} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \boldsymbol{\Omega}\right)^\top \boldsymbol{y} \end{bmatrix}
\end{aligned}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{g}_{1:n+1} & \boldsymbol{g}_{2:n+1} & \cdots & \boldsymbol{g}_{n:n+1} \end{bmatrix} \in \mathbb{R}^{n\times d}.$$

Then if taking the last entry of $\boldsymbol{o}_{n+1}$ as final prediction, we get

$$f_{\mathsf{GLA}}(\boldsymbol{Z}) = \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \boldsymbol{\Omega}\right)^\top \boldsymbol{y}$$

which completes the proof of Theorem 1. □

Construction 2: Based on the construction given in (26), only d elements of $\boldsymbol{G}_i$ matrices are useful. One might ask about the effect of other entries of $\boldsymbol{G}_i$. Therefore, in the following, we introduce an other model construction showing that different row of $\boldsymbol{G}_i$ implements WPGD with different weighting. Similarly, let $\boldsymbol{W}_k, \boldsymbol{W}_q$ be the same as (26) but with $\boldsymbol{W}_v$ constructed by

$$\boldsymbol{W}_v = \begin{bmatrix} \boldsymbol{0}_{(d+1)\times d} & \boldsymbol{u} \end{bmatrix}^\top \quad \text{where} \quad \boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_{d+1} \end{bmatrix}^\top \in \mathbb{R}^{d+1}.$$

Then, the value embeddings have the form of $\boldsymbol{v}_i = y_i \boldsymbol{u}$, which gives

$$\boldsymbol{v}_i \boldsymbol{k}_i^\top = \boldsymbol{u} \begin{bmatrix} y_i \boldsymbol{x}_i^\top \boldsymbol{P}_k & \boldsymbol{0} \end{bmatrix}.$$

Next, let

$$\boldsymbol{G}_i = \begin{bmatrix} (\boldsymbol{g}_i^1)^\top & * \\ (\boldsymbol{g}_i^2)^\top & * \\ \vdots & \vdots \\ (\boldsymbol{g}_i^{d+1})^\top & * \end{bmatrix}, \quad \text{and} \quad \boldsymbol{G}_{j:i} = \begin{bmatrix} (\boldsymbol{g}_{j:i}^1)^\top & * \\ (\boldsymbol{g}_{j:i}^2)^\top & * \\ \vdots & \vdots \\ (\boldsymbol{g}_{j:i}^{d+1})^\top & * \end{bmatrix},$$

where $\boldsymbol{g}_i^{i'} \in \mathbb{R}^d$ corresponds to the i'-th row of $\boldsymbol{G}_i$ and $\boldsymbol{g}_{j:i}^{i'} = \boldsymbol{g}_{j+1}^{i'} \odot \boldsymbol{g}_{j+1}^{i'} \cdots \boldsymbol{g}_i^{i'}$. Then we get the output

$$\boldsymbol{o}_{n+1} = \begin{bmatrix} \sum_{j=1}^{n} u_1 y_j \boldsymbol{x}_j^\top \boldsymbol{P}_k \odot (\boldsymbol{g}_{j:n+1}^1)^\top & 0 \\ \sum_{j=1}^{n} u_2 y_j \boldsymbol{x}_j^\top \boldsymbol{P}_k \odot (\boldsymbol{g}_{j:n+1}^2)^\top & 0 \\ \vdots & \\ \sum_{j=1}^{n} u_{d+1} y_j \boldsymbol{x}_j^\top \boldsymbol{P}_k \odot (\boldsymbol{g}_{j:n+1}^{d+1})^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_q^\top \boldsymbol{x} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \boldsymbol{\Omega}_1\right)^\top \boldsymbol{y} \\ \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \boldsymbol{\Omega}_2\right)^\top \boldsymbol{y} \\ \vdots \\ \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \boldsymbol{\Omega}_{d+1}\right)^\top \boldsymbol{y} \end{bmatrix},$$

where

$$\boldsymbol{\Omega}_i = u_i \begin{bmatrix} \boldsymbol{g}_{1:n+1}^i & \boldsymbol{g}_{2:n+1}^i & \cdots & \boldsymbol{g}_{n:n+1}^i \end{bmatrix} \in \mathbb{R}^{n\times d}, \quad i \le d+1. \tag{27}$$

Therefore, consider $(d+1)$-dimensional output $\boldsymbol{o}_{n+1}$. Each entry implements a 1-step WPGD with same preconditioners $\boldsymbol{P}_k, \boldsymbol{P}_q$ and different weighting matrices $\boldsymbol{\Omega}$'s. The weighting matrix of i'th entry is determined by the i'th row of all gating matrices. Note that if consider the last entry of $\boldsymbol{o}_{n+1}$ as prediction, it returns the same result as **Construction 1** above, where only last rows of $\boldsymbol{G}_i$'s are useful.

Additionally, suppose that the final prediction is given after a linear head $\boldsymbol{h}$, that is, $f_{\mathsf{GLA}}(\boldsymbol{Z}) = \boldsymbol{h}^\top \boldsymbol{o}_{n+1}$, and let $\boldsymbol{h} = \begin{bmatrix} h_1 & h_2 & \cdots & h_{d+1} \end{bmatrix}^\top \in \mathbb{R}^{d+1}$. Then

$$f_{\mathsf{GLA}}(\boldsymbol{Z}) = \boldsymbol{h}^\top \boldsymbol{o}_{n+1} = \boldsymbol{x}^\top \boldsymbol{P}_q \left(\boldsymbol{X}\boldsymbol{P}_k \odot \bar{\boldsymbol{\Omega}}\right)^\top \boldsymbol{y}, \tag{28}$$

where

$$\bar{\mathbf{\Omega}} = \sum_{i=1}^{d+1} h_i \mathbf{\Omega}_i = \sum_{i=1}^{d+1} h_i u_i \begin{bmatrix} \boldsymbol{g}^i_{1:n+1} & \boldsymbol{g}^i_{2:n+1} & \cdots & \boldsymbol{g}^i_{n:n+1} \end{bmatrix} \in \mathbb{R}^{n \times d}. \tag{29}$$

Then, single-layer GLA still returns one-step WPGD with updated weighting matrix.

D.2 Proof of Theorem 6

In this section, we first prove the following theorem, which differs from Theorem 6 only in that $\boldsymbol{W}_{q,\ell}$ is parameterized by $\boldsymbol{P}_{q,\ell}$ instead of $-\boldsymbol{P}_{q,\ell}$. Theorem 6 can then be directly obtained by setting $\boldsymbol{P}_{q,\ell}$ to $-\boldsymbol{P}_{q,\ell}$.

Theorem 7. *Consider an L-layer GLA with residual connections, where each layer follows the weight construction specified in (9). Let $\boldsymbol{W}_{k,\ell}$ and $\boldsymbol{W}_{q,\ell}$ in the ℓ'th layer parameterized by $\boldsymbol{P}_{k,\ell}, \boldsymbol{P}_{q,\ell} \in \mathbb{R}^{d\times d}$, for $\ell \in [L]$. Let the gating be a function of the features, e.g., $\boldsymbol{G}_i = G(\boldsymbol{x}_i)$, and define $\mathbf{\Omega}$ as in Theorem 1. Additionally, denote the masking as $\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{I}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{n\times n}$, and let $\hat{\boldsymbol{\beta}}_0 = \boldsymbol{\beta}_{i,0} = \boldsymbol{0}$ for all $i \in [n]$.*

Then, the last entry of the i'th token output at the ℓ'th layer ($o_{i,\ell}$) is:

- *For all $i \le n$, $o_{i,\ell} = y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_{i,\ell}$, where $\boldsymbol{\beta}_{i,\ell} = \boldsymbol{\beta}_{i,\ell-1} + \boldsymbol{P}_{q,\ell}\left(\nabla_{i,\ell} \oslash \boldsymbol{g}_{i:n+1}\right)$,*
- *$o_{n+1,\ell} = -\boldsymbol{x}^\top \hat{\boldsymbol{\beta}}_\ell$, where $\hat{\boldsymbol{\beta}}_\ell = (1+\alpha_\ell)\hat{\boldsymbol{\beta}}_{\ell-1} + \boldsymbol{P}_{q,\ell}\left(\nabla_{n,\ell} \oslash \boldsymbol{g}_{n+1}\right)$, and $\alpha_\ell = \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell}\boldsymbol{P}_{k,\ell}^\top \boldsymbol{x}$.*

Here, letting $\boldsymbol{B}_\ell = [\boldsymbol{\beta}_{1,\ell} \ \cdots \ \boldsymbol{\beta}_{n,\ell}]^\top$, $\boldsymbol{X}_\ell = \boldsymbol{X}\boldsymbol{P}_{k,\ell} \odot \mathbf{\Omega}$, and $\boldsymbol{y}_\ell = \mathrm{diag}(\boldsymbol{X}\boldsymbol{B}_\ell^\top)$, we define

$$\nabla_{i,\ell} = \boldsymbol{X}_\ell^\top \boldsymbol{M}_i \left(\boldsymbol{y}_\ell - \boldsymbol{y}\right).$$

Proof. Suppose that the input prompt for ℓ'th layer is

$$\boldsymbol{Z}_\ell = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 & \cdots & \tilde{\boldsymbol{x}}_n & \tilde{\boldsymbol{x}}_{n+1} \\ \tilde{y}_1 & \cdots & \tilde{y}_n & \tilde{y}_{n+1} \end{bmatrix}^\top \in \mathbb{R}^{(n+1)\times(d+1)} \tag{30}$$

and let $\boldsymbol{k}_{i,\ell}, \boldsymbol{q}_{i,\ell}, \boldsymbol{v}_{i,\ell}$ be their corresponding key, query and value embeddings. Recap the model construction from (9) and follow the same analysis in Appendix D.1. Let $\boldsymbol{S}_i^\ell, i \in [n+1]$ be the corresponding states. We have

$$\begin{aligned} \boldsymbol{S}_i^\ell &= \sum_{j=1}^{i} \boldsymbol{G}_{j:i}^\ell \odot \boldsymbol{v}_{j,\ell}\boldsymbol{k}_{j,\ell}^\top \\ &= \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \sum_{j=1}^{i} \tilde{y}_j \tilde{\boldsymbol{x}}_j^\top \boldsymbol{P}_{k,\ell} \odot (\boldsymbol{g}_{j:i}^\ell)^\top & 0 \end{bmatrix} \end{aligned}$$

where

$$\boldsymbol{G}_i^\ell = G(\tilde{\boldsymbol{x}}_i) = \begin{bmatrix} * & * \\ (\boldsymbol{g}_i^\ell)^\top & * \end{bmatrix} \quad \text{and} \quad \boldsymbol{G}_{j:i}^\ell = \begin{bmatrix} * & * \\ (\boldsymbol{g}_{j:i}^\ell)^\top & * \end{bmatrix}.$$

Additionally, recap that we have $\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{I}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$ and $\oslash$ denotes Hadamard division. Let

$$\mathbf{\Omega}^\ell = \begin{bmatrix} \boldsymbol{g}_{i:n+1}^\ell & \cdots & \boldsymbol{g}_{n:n+1}^\ell \end{bmatrix}.$$

Then, defining

$$\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1 \ \tilde{\boldsymbol{x}}_2 \ \cdots \ \tilde{\boldsymbol{x}}_n]^\top \in \mathbb{R}^{n\times d} \quad \text{and} \quad \tilde{\boldsymbol{y}} = [\tilde{y}_1 \ \tilde{y}_2 \ \cdots \ \tilde{y}_n]^\top \in \mathbb{R}^n,$$

and letting $\boldsymbol{o}_{i,\ell}, i \in [n+1]$ be their outputs, we get:

- For $i \le n$,

$$\sum_{j=1}^{i} \tilde{y}_j \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_j \odot \boldsymbol{g}_{j:i}^\ell = \left(\sum_{j=1}^{i} \tilde{y}_j \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_j \odot \boldsymbol{g}_{j:n+1}^\ell \right) \oslash \boldsymbol{g}_{i:n+1}^\ell$$
$$= (\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \boldsymbol{M}_i \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{i:n+1}^\ell,$$

and

$$\boldsymbol{o}_{i,\ell} = \boldsymbol{S}_i^\ell \boldsymbol{q}_{i,\ell} = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \left((\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \boldsymbol{M}_i \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{i:n+1}^\ell \right)^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{q,\ell}^\top \tilde{\boldsymbol{x}}_i \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{x}}_i^\top \boldsymbol{P}_{q,\ell} \left((\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \boldsymbol{M}_i \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{i:n+1}^\ell \right) \end{bmatrix}. \tag{31}$$

- For $i = n+1$,

$$\sum_{j=1}^{n+1} \tilde{y}_j \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_j \odot \boldsymbol{g}_{j:i}^\ell = (\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \tilde{\boldsymbol{y}} + \tilde{y}_{n+1} \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_{n+1},$$

and

$$\boldsymbol{o}_{n+1,\ell} = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0} \\ \left((\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \tilde{\boldsymbol{y}} + \tilde{y}_{n+1} \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_{n+1} \right)^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{q,\ell}^\top \tilde{\boldsymbol{x}}_{n+1} \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{x}}_{n+1}^\top \boldsymbol{P}_{q,\ell} \left((\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \tilde{\boldsymbol{y}} + \tilde{y}_{n+1} \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_{n+1} \right) \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{0} \\ \tilde{y}_{n+1} \tilde{\boldsymbol{x}}_{n+1}^\top \boldsymbol{P}_{q,\ell} \boldsymbol{P}_{k,\ell}^\top \tilde{\boldsymbol{x}}_{n+1} + \tilde{\boldsymbol{x}}_{n+1}^\top \boldsymbol{P}_{q,\ell} \left((\tilde{\boldsymbol{X}} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}^\ell)^\top \tilde{\boldsymbol{y}} \right). \end{bmatrix} \tag{32}$$

From (31) and (32), the first d entries of all the $(d+1)$-dimensional outputs (e.g, $\boldsymbol{o}_{i,\ell}$, $i \in [n+1]$) are zero. Given the multi-layer GLA as formulated in (25) where residual connections are applied, we have that the first d dimension of all the input tokens remain the same. That is, $\tilde{\boldsymbol{x}}_i \equiv \boldsymbol{x}_i$. Additionally, since gating only depends on the feature $\boldsymbol{x}_i$, then all the layers return the same gatings (e.g, $\boldsymbol{G}_i^\ell \equiv \boldsymbol{G}_i$).

Note that if the gating function varies across layers or depends on the entire token rather than just the feature $\boldsymbol{x}_i$, each layer will have its own gating, leading to $\boldsymbol{G}_i^\ell \neq \boldsymbol{G}_i$. In this theorem, we assume identical gating matrices across layers for simplicity and clarity. However, an extended version of this theorem, without this assumption, can be directly derived.

Therefore, we obtain

$$\boldsymbol{o}_{i,\ell} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{i:n+1} \right) \end{bmatrix}, \quad i \le n$$
$$\boldsymbol{o}_{n+1,\ell} = \begin{bmatrix} \boldsymbol{0} \\ \tilde{y}_{n+1} \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \boldsymbol{P}_{k,\ell}^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{n+1} \right) \end{bmatrix}.$$

where we define $\boldsymbol{X}_\ell := \boldsymbol{X} \boldsymbol{P}_{k,\ell} \odot \boldsymbol{\Omega}$.

Since we focus on the last/$(d+1)$'th entry of each token output, we have

$$o_{i,\ell} = \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{i:n+1} \right), \quad i \le n,$$
$$o_{n+1,\ell} = \tilde{y}_{n+1} \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \boldsymbol{P}_{k,\ell}^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \tilde{\boldsymbol{y}} \oslash \boldsymbol{g}_{n+1} \right). \tag{33}$$

It remains to get the $\tilde{y}_i$ for each layer's input. Let inputs of $(\ell-1)$'th and ℓ'th layer be

$$\boldsymbol{Z}_{\ell-1} = \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n & \boldsymbol{x} \\ y_1^{\ell-1} & \dots & y_n^{\ell-1} & y^{\ell-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{Z}_\ell = \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n & \boldsymbol{x} \\ y_1^{\ell} & \dots & y_n^{\ell} & y^{\ell} \end{bmatrix}.$$

Define

$$\boldsymbol{y}^{\ell-1} := [y_1^{\ell-1} \ \cdots \ y_n^{\ell-1}]^\top \in \mathbb{R}^n.$$

- For $i \leq n$, from (33) and (25), we have

$$y_i^\ell = y_i^{\ell-1} + \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i \boldsymbol{y}^{\ell-1} \oslash \boldsymbol{g}_{i:n+1} \right)$$
$$\implies y_i^\ell - y_i^{\ell-1} = \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i \boldsymbol{y}^{\ell-1} \oslash \boldsymbol{g}_{i:n+1} \right).$$

  Suppose that $y_i - y_i^\ell = \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^\ell$. Then

$$\boldsymbol{y}^\ell = \boldsymbol{y} - [\boldsymbol{x}_1^\top \boldsymbol{\beta}_1^\ell \ \cdots \ \boldsymbol{x}_n^\top \boldsymbol{\beta}_n^\ell]^\top = \boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top) \tag{34}$$

  and

$$y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^\ell - (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^{\ell-1}) = \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{i:n+1} \right)$$
$$\implies \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^\ell = \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^{\ell-1} - \boldsymbol{x}_i^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{i:n+1} \right).$$

  Hence,

$$y_i^\ell = y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_i^\ell \quad \text{where} \quad \boldsymbol{\beta}_i^\ell = \boldsymbol{\beta}_i^{\ell-1} - \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{M}_i (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{i:n+1} \right) \tag{35}$$

- For $i = n+1$, from (33) and (25), we have

$$y^\ell = y^{\ell-1} + y^{\ell-1} \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \boldsymbol{P}_{k,\ell}^\top \boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top \boldsymbol{y}^{\ell-1} \oslash \boldsymbol{g}_{n+1} \right).$$

  Setting $\alpha_\ell = \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \boldsymbol{P}_{k,\ell}^\top \boldsymbol{x}$ and recalling $\boldsymbol{y}^\ell$ from (34), we get

$$y^\ell = (1 + \alpha_\ell) y^{\ell-1} + \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{n+1} \right).$$

  Additionally, let $y^\ell = -\boldsymbol{x}^\top \boldsymbol{\beta}^\ell$. Then,

$$-\boldsymbol{x}^\top \boldsymbol{\beta}^\ell = -(1 + \alpha_\ell) \boldsymbol{x}^\top \boldsymbol{\beta}^{\ell-1} + \boldsymbol{x}^\top \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{n+1} \right).$$

  Hence,

$$y^\ell = -\boldsymbol{x}^\top \boldsymbol{\beta}^\ell \quad \text{where} \quad \boldsymbol{\beta}^\ell = (1 + \alpha_\ell) \boldsymbol{\beta}^{\ell-1} - \boldsymbol{P}_{q,\ell} \left(\boldsymbol{X}_\ell^\top (\boldsymbol{y} - \operatorname{diag}(\boldsymbol{X}\boldsymbol{B}^\top)) \oslash \boldsymbol{g}_{n+1} \right) \tag{36}$$

Combining (35) and (36) together completes the proof. □

E Optimization landscape of WPGD

We first provide the following Lemma.

Lemma 1 (Existence of Fixed Point)**.** *Consider the functions $h_1(\bar{\gamma})$ and $h_2(\gamma)$ defined in (14a) and (14b), respectively. The composite function $h_1(h_2(\gamma)) + 1$ has at least one fixed point $\gamma^\star \geq 1$, i.e., there exists $\gamma^\star \geq 1$ such that $h_1(h_2(\gamma^\star)) + 1 = \gamma^\star$.*

Proof. We prove existence using continuity and boundedness arguments along with the intermediate value theorem. Note that $h_1(\bar{\gamma})$ and $h_2(\gamma)$ are continuous functions. Thus, their composition $f(\gamma) := h_1(h_2(\gamma)) + 1$ is continuous for all $\gamma \geq 1$.

We examine the behavior of $f(\gamma)$ at the boundaries of its domain: At $\gamma = 1$, we have $h_2(1) = c_0 > 0$, where c_0 is a positive constant. Therefore, $f(1) = h_1(h_2(1)) + 1 = h_1(c_0) + 1 > 0$, which is finite and positive since h_1 is positive for all non-negative inputs. As $\gamma \to \infty$, $h_2(\gamma) \to c_\infty > 0$, where c_∞ is a positive constant.

For sufficiently large γ, we can establish that $f(\gamma) < \gamma$. To see this, note that we have shown that $h_2(\gamma) \to c_\infty$ as $\gamma \to \infty$, where c_∞ is a fixed positive constant. Further, for any fixed positive value $\bar{\gamma}$, the function $h_1(\bar{\gamma})$ is bounded, i.e., $h_1(\bar{\gamma}) \leq \max_{1 \leq i \leq n} \lambda_i$, which follows

from the fact that $h_1(\bar{\gamma})$ is a weighted average of the λ_i values. Therefore, there exists a constant K such that $f(\gamma) \leq K$ for all $\gamma \geq 1$. Hence, it follows that for all $\gamma > K$, we have $f(\gamma) \leq K < \gamma$.

Now, consider the function $\bar{f}(\gamma) = f(\gamma) - \gamma$, which is continuous on $[1, \infty)$ since both $f(\gamma)$ and γ are continuous. We have $\bar{f}(1) = f(1) - 1 = h_1(c_0) > 0$ and for some $\gamma_1 > K$, we have $\bar{f}(\gamma_1) = f(\gamma_1) - \gamma_1 < 0$. Since $\bar{f}$ is continuous and changes sign from positive to negative on the interval $[1, \gamma_1]$, by the Intermediate Value Theorem, there exists at least one $\gamma^\star \in (1, \gamma_1)$ such that $\bar{f}(\gamma^\star) = 0$. This implies $f(\gamma^\star) - \gamma^\star = 0$, or equivalently, $f(\gamma^\star) = \gamma^\star$. Therefore, there exists at least one fixed point $\gamma^\star \geq 1$ for the function $h_1(h_2(\gamma)) + 1$. □

E.1 Proof of Theorem 2

Proof. Recapping the objective from (2) and following Definitions 1 and 2, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{P}, \boldsymbol{\omega}) &= \mathbb{E}\left[\left(y - \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right)^2\right] \\ &= \mathbb{E}\left[y^2\right] - 2\mathbb{E}\left[y\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right] + \mathbb{E}\left[\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right)^2\right].\end{aligned}$$

Let $y = \boldsymbol{x}^\top \boldsymbol{\beta} + \xi$ and $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i$, for all $i \in [n]$, where $\xi, \xi_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. Then,

$$\mathbb{E}[y^2] = \mathbb{E}[(\boldsymbol{x}^\top \boldsymbol{\beta} + \xi)^2] = \mathtt{tr}\left(\boldsymbol{\Sigma}\right) + \sigma^2,$$

and

$$\begin{aligned}\mathbb{E}\left[y\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right] &= \mathbb{E}\left[(\boldsymbol{\beta}^\top \boldsymbol{x} + \xi)\boldsymbol{x}^\top \boldsymbol{P}\sum_{i=1}^n \omega_i \boldsymbol{x}_i(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)\right] \\ &= \mathbb{E}\left[\boldsymbol{\beta}^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{P}\sum_{i=1}^n \omega_i \boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{\beta}_i\right] \\ &= \mathtt{tr}\left(\boldsymbol{\Sigma}\boldsymbol{P}\boldsymbol{\Sigma}\sum_{i=1}^n \omega_i \mathbb{E}\left[\boldsymbol{\beta}_i\boldsymbol{\beta}^\top\right]\right) \\ &= \mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}\right)\boldsymbol{\omega}^\top \boldsymbol{r}.\end{aligned}$$

Here, the last equality comes from the fact that since $\boldsymbol{\beta}_i - r_{ij}\boldsymbol{\beta}_j$ is independent of $\boldsymbol{\beta}_j$ for $i, j \in [n+1]$ following Definition 1, we have $\mathbb{E}\left[\boldsymbol{\beta}_i\boldsymbol{\beta}^\top\right] = r_{i,n+1}\boldsymbol{I}_d$ and $\sum_{i=1}^n \omega_i \mathbb{E}\left[\boldsymbol{\beta}_i\boldsymbol{\beta}^\top\right]$ returns $\boldsymbol{\omega}^\top \boldsymbol{r} \cdot \boldsymbol{I}_d$.

Hence,

$$\begin{aligned}\mathbb{E}\left[\left(\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{X}(\boldsymbol{\omega} \odot \boldsymbol{y})\right)^2\right] &= \mathbb{E}\left[\boldsymbol{x}^\top \boldsymbol{P}\left(\sum_{i=1}^n \omega_i(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)\boldsymbol{x}_i\right)\left(\sum_{i=1}^n \omega_i \boldsymbol{x}_i^\top(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)\right)\boldsymbol{P}^\top \boldsymbol{x}\right] \\ &= \mathtt{tr}\left(\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i=1}^n \omega_i^2(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)^2 \boldsymbol{x}_i\boldsymbol{x}_i^\top\right]\right) \\ &\quad + \mathtt{tr}\left(\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i\neq j} \omega_i\omega_j(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)\boldsymbol{x}_i\boldsymbol{x}_j^\top(\boldsymbol{x}_j^\top \boldsymbol{\beta}_j + \xi_j)\right]\right),\end{aligned}$$

where

$$\begin{aligned}\mathtt{tr}\left(\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i=1}^n \omega_i^2(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i + \xi_i)^2 \boldsymbol{x}_i\boldsymbol{x}_i^\top\right]\right) &= \mathtt{tr}\left(\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i=1}^n \omega_i^2(\boldsymbol{x}_i^\top \boldsymbol{\beta}_i\boldsymbol{\beta}_i^\top \boldsymbol{x}_i + \sigma^2)\boldsymbol{x}_i\boldsymbol{x}_i^\top\right]\right) \\ &= \|\boldsymbol{\omega}\|_{\ell_2}^2 \,\mathtt{tr}\left(\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{x}\boldsymbol{x}^\top\right] + \sigma^2\boldsymbol{\Sigma}\right)\right) \\ &= \|\boldsymbol{\omega}\|_{\ell_2}^2 \left(\mathtt{tr}\left(\boldsymbol{\Sigma}\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\right)\left(\mathtt{tr}\left(\boldsymbol{\Sigma}\right) + \sigma^2\right) + 2\mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}^\top \boldsymbol{\Sigma}\boldsymbol{P}\right)\right),\end{aligned}$$

and

$$\begin{aligned}\mathtt{tr}\left(\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i\neq j}\omega_i\omega_j(\boldsymbol{x}_i^\top\boldsymbol{\beta}_i+\xi_i)\boldsymbol{x}_i\boldsymbol{x}_j^\top(\boldsymbol{x}_j^\top\boldsymbol{\beta}_j+\xi_j)\right]\right) &= \mathtt{tr}\left(\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\mathbb{E}\left[\sum_{i\neq j}\omega_i\omega_j\boldsymbol{x}_i\boldsymbol{x}_i^\top\boldsymbol{\beta}_i\boldsymbol{\beta}_j^\top\boldsymbol{x}_j\boldsymbol{x}_j^\top\right]\right)\\ &= \mathtt{tr}\left(\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right)\sum_{i\neq j}\omega_i\omega_j\mathbb{E}[\boldsymbol{\beta}_i^\top\boldsymbol{\beta}_j]\\ &= \mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right)\boldsymbol{\omega}^\top(\boldsymbol{R}-\boldsymbol{I})\boldsymbol{\omega}.\end{aligned}$$

Combining all together and letting $M := \mathtt{tr}\,(\boldsymbol{\Sigma})+\sigma^2$, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{P},\boldsymbol{\omega}) = M &- 2\mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}\right)\boldsymbol{\omega}^\top\boldsymbol{r}\\ &+ M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2\mathtt{tr}\left(\boldsymbol{\Sigma}\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right)+(\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega})\mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right).\end{aligned}\tag{37}$$

For simplicity, and without loss of generality, let

$$\tilde{\boldsymbol{P}} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{P}\sqrt{\boldsymbol{\Sigma}}.\tag{38}$$

Then, we obtain

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{P}},\boldsymbol{\omega}) = M &- 2\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)\boldsymbol{\omega}^\top\boldsymbol{r}\\ &+ M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+(\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega})\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right).\end{aligned}\tag{39}$$

Further, the gradients can be written as

$$\nabla_{\tilde{\boldsymbol{P}}}\mathcal{L}(\tilde{\boldsymbol{P}},\boldsymbol{\omega}) = -2\boldsymbol{\omega}^\top\boldsymbol{r}\boldsymbol{\Sigma}+2M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2\tilde{\boldsymbol{P}}+2(\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega})\boldsymbol{\Sigma}\tilde{\boldsymbol{P}},\tag{40}$$

$$\nabla_{\boldsymbol{\omega}}\mathcal{L}(\tilde{\boldsymbol{P}},\boldsymbol{\omega}) = -2\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)\boldsymbol{r}+2M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)\boldsymbol{\omega}+2\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)(\boldsymbol{I}_n+\boldsymbol{R})\boldsymbol{\omega}.\tag{41}$$

Using the first-order optimality condition, and setting $\nabla_{\tilde{\boldsymbol{P}}}\mathcal{L}(\tilde{\boldsymbol{P}},\boldsymbol{\omega})=0$ and $\nabla_{\boldsymbol{\omega}}\mathcal{L}(\tilde{\boldsymbol{P}},\boldsymbol{\omega})=0$, we obtain

$$\begin{aligned}\tilde{\boldsymbol{P}} &= \left(M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2\boldsymbol{I}+(\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega})\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\Sigma}\boldsymbol{\omega}^\top\boldsymbol{r}\\ &= \frac{\boldsymbol{\omega}^\top\boldsymbol{r}}{M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2}\left(\frac{\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega}}{M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2}\boldsymbol{I}+\boldsymbol{\Sigma}^{-1}\right)^{-1}\\ &= \frac{\boldsymbol{\omega}^\top\boldsymbol{r}}{M\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2}\left(\frac{\gamma}{M}\cdot\boldsymbol{I}+\boldsymbol{\Sigma}^{-1}\right)^{-1},\end{aligned}\tag{42a}$$

where

$$\gamma := \frac{\boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega}}{\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2}+1.$$

Further,

$$\begin{aligned}\boldsymbol{\omega} &= \left(\left(M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)\right)\boldsymbol{I}+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)\boldsymbol{R}\right)^{-1}\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)\boldsymbol{r}\\ &= \frac{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)}{M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}\left(\boldsymbol{I}+\frac{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}{M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}\boldsymbol{R}\right)^{-1}\boldsymbol{r}.\end{aligned}\tag{42b}$$

Let

$$\boldsymbol{\Sigma}_\gamma := \frac{\gamma}{M}\cdot\boldsymbol{I}+\boldsymbol{\Sigma}^{-1}.$$

Then, we get

$$\begin{aligned}\frac{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}{M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)} &= \left(1+M\frac{\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}\right)^{-1}\\ &= \left(1+M\frac{\mathtt{tr}\left(\boldsymbol{\Sigma}_\gamma^{-2}\right)}{\mathtt{tr}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_\gamma^{-2}\right)}\right)^{-1}\\ &= \left(1+M\sum_{i=1}^d\frac{s_i^2}{(M+\gamma s_i)^2}\left(\sum_{i=1}^d\frac{s_i^3}{(M+\gamma s_i)^2}\right)^{-1}\right)^{-1}\\ &=: h_2(\gamma).\end{aligned}$$

Here, the last equality follows from eigen decomposition $\boldsymbol{\Sigma} = \boldsymbol{U}\mathrm{diag}(\mathbf{s})\boldsymbol{U}^\top$ with $\boldsymbol{s} = [s_1,\ldots,s_d]^\top \in \mathbb{R}^d_{++}$.

Now, plugging $\tilde{\boldsymbol{P}}$ defined in (42a) within $\boldsymbol{\omega}$ given in (42b), we obtain

$$\boldsymbol{\omega} = \frac{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)}{M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}\cdot\left(h_2(\gamma)\cdot\boldsymbol{R}+\boldsymbol{I}\right)^{-1}\boldsymbol{r}. \tag{43}$$

Using the above formulae for $\boldsymbol{\omega}$, we rewrite $\gamma = \boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega}/\left\|\boldsymbol{\omega}\right\|_{\ell_2}^2+1$ as

$$\begin{aligned}\gamma-1 &= \frac{\boldsymbol{r}^\top(h_2(\gamma)\boldsymbol{R}+\boldsymbol{I})^{-1}\boldsymbol{R}(h_2(\gamma)\boldsymbol{R}+\boldsymbol{I})^{-1}\boldsymbol{r}}{\boldsymbol{r}^\top(h_2(\gamma)\boldsymbol{R}+\boldsymbol{I})^{-2}\boldsymbol{r}}\\ &= \sum_{i=1}^n\frac{\lambda_i a_i^2}{\left(1+h_2(\gamma)\lambda_i\right)^2}\left(\sum_{i=1}^n\frac{a_i^2}{\left(1+h_2(\gamma)\lambda_i\right)^2}\right)^{-1}\\ &=: h_1(h_2(\gamma)),\end{aligned} \tag{44}$$

where the second equality follows from Assumption 1 where $\boldsymbol{r} = \boldsymbol{E}\boldsymbol{a}$ with $\boldsymbol{a} = [a_1,\cdots,a_n]^\top \in \mathbb{R}^n$, and the fact that $\boldsymbol{R} = \boldsymbol{E}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{E}^\top$ denotes the eigen decomposition of $\boldsymbol{R}$, with $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_n]^\top \in \mathbb{R}^n_+$.

It follows from Lemma 1 that there exists $\gamma^\star \geq 1$ such that $h_1(h_2(\gamma^\star))+1 = \gamma^\star$. From (42a) and (43), we obtain

$$\begin{aligned}\tilde{\boldsymbol{P}} &= C(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma})\cdot\left(\frac{\gamma^\star}{M}\cdot\boldsymbol{I}+\boldsymbol{\Sigma}^{-1}\right)^{-1}, \quad \text{and}\\ \boldsymbol{\omega} &= c(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma})\cdot\left(h_2(\gamma^\star)\cdot\boldsymbol{R}+\boldsymbol{I}\right)^{-1}\boldsymbol{r}.\end{aligned} \tag{45}$$

for some $C(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma}) = \frac{\boldsymbol{\omega}^\top\boldsymbol{r}}{M\|\boldsymbol{\omega}\|_{\ell_2}^2}$ and $c(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma}) = \frac{\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}\right)}{M\mathtt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)+\mathtt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)}$.

Now, using the our definition $\tilde{\boldsymbol{P}} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{P}\sqrt{\boldsymbol{\Sigma}}$, we obtain

$$\begin{aligned}\boldsymbol{P}(\gamma^\star) &= C(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma})\cdot\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\frac{\gamma^\star}{M}\cdot\boldsymbol{\Sigma}+\boldsymbol{I}\right)^{-1}\boldsymbol{\Sigma}^{-\frac{1}{2}}, \quad \text{and}\\ \boldsymbol{\omega}(\gamma^\star) &= c(\boldsymbol{r},\boldsymbol{\omega},\boldsymbol{\Sigma})\cdot\left(h_2(\gamma^\star)\cdot\boldsymbol{R}+\boldsymbol{I}\right)^{-1}\boldsymbol{r}.\end{aligned}$$

This completes the proof. □

E.2 Proof of Theorem 3

We first provide the following Lemma.

Lemma 2. *Consider the functions $h_1(\bar{\gamma})$ and $h_2(\gamma)$ defined in (14a) and (14b), respectively, where $\lambda_i \geq 0$, $a_i \neq 0$ for at least one i, $s_i > 0$, and $M = \sigma^2 + \sum_{i=1}^d s_i > 0$. Suppose $\Delta_{\boldsymbol{\Sigma}} \cdot \Delta_{\boldsymbol{R}} < M + s_{\min}$, where $\Delta_{\boldsymbol{\Sigma}}$ and $\Delta_{\boldsymbol{R}}$ denote the effective spectral gaps of $\boldsymbol{\Sigma}$ and $\boldsymbol{R}$, respectively, as given in (13); and $s_{\min}$ is the smallest eigenvalue of $\boldsymbol{\Sigma}$. We have that*

$$\left|\frac{\partial h_2}{\partial \gamma} \cdot \frac{\partial h_1}{\partial h_2}\right| \leq \frac{\Delta_{\boldsymbol{\Sigma}}^2 \cdot \Delta_{\boldsymbol{R}}^2}{(M + s_{\min})^2} < 1.$$

Proof. Let

$$B(\gamma) = \sum_{i=1}^d \frac{s_i^3}{(M + \gamma s_i)^2}, \quad C(\gamma) = \sum_{i=1}^d \frac{s_i^2}{(M + \gamma s_i)^2}, \quad A(\gamma) = 1 + M\frac{C(\gamma)}{B(\gamma)}.$$

The derivatives of $B(\gamma)$ and $C(\gamma)$ are

$$B'(\gamma) = -2\sum_{i=1}^d \frac{s_i^4}{(M + \gamma s_i)^3}, \quad C'(\gamma) = -2\sum_{i=1}^d \frac{s_i^3}{(M + \gamma s_i)^3}.$$

The gradient of $h_2(\gamma) = A(\gamma)^{-1}$ is

$$\frac{\partial h_2}{\partial \gamma} = -M\left(\frac{1}{A(\gamma)B(\gamma)}\right)^2 \left(C'(\gamma)B(\gamma) - C(\gamma)B'(\gamma)\right). \tag{46}$$

It can be seen that

$$\left(\frac{1}{A(\gamma)}\right)^2 \leq M^{-2}\left(\sum_{i=1}^d \frac{s_i^3}{(M + \gamma s_i)^2}\right)^2 \left(\sum_{i=1}^d \frac{s_i^2}{(M + \gamma s_i)^2}\right)^{-2},$$

which implies that

$$\left(\frac{1}{A(\gamma)B(\gamma)}\right)^2 \leq M^{-2}\left(\sum_{i=1}^d \frac{s_i^2}{(M + \gamma s_i)^2}\right)^{-2}. \tag{47a}$$

Further, we have

$$\begin{aligned} C'(\gamma)B(\gamma) - C(\gamma)B'(\gamma) &= -2\sum_{i=1}^d \frac{s_i^3}{(M+\gamma s_i)^3}\sum_{i=1}^d \frac{s_i^3}{(M+\gamma s_i)^2} \\ &\quad + 2\sum_{i=1}^d \frac{s_i^2}{(M+\gamma s_i)^2}\sum_{i=1}^d \frac{s_i^4}{(M+\gamma s_i)^3} \\ &= \sum_{i=1}^d\sum_{j=1}^d \frac{T_{ij}}{(M+\gamma s_i)^3(M+\gamma s_j)^3} \\ &= M \cdot \sum_{i=1}^d\sum_{j=1}^d \frac{s_i^2 s_j^2 (s_i - s_i)^2}{(M+\gamma s_i)^3(M+\gamma s_j)^3}, \end{aligned} \tag{47b}$$

where

$$\begin{aligned} T_{ij} &= s_i^2(M+\gamma s_i)s_j^4 + s_i^4 s_j^2(M+\gamma s_j) \\ &\quad - s_i^3 s_j^3(M+\gamma s_j) - s_i^3(M+\gamma s_i)s_j^3 \\ &= s_i^2 s_j^2\left(M\cdot(s_j^2 + s_i^2 - 2s_i s_j) + \gamma(s_i s_j^2 + s_i^2 s_j - s_i s_j^2 - s_i^2 s_j)\right). \end{aligned} \tag{47c}$$

Thus, substituting (47a) and (47b) into (46), we obtain

$$\left|\frac{\partial h_2}{\partial \gamma}\right| \leq M \cdot M^{-1}\left(\sum_{i=1}^d \frac{s_i^2}{(M+\gamma s_i)^2}\right)^{-2} \sum_{i,j=1}^d \frac{s_i^2 s_j^2 (s_i - s_j)^2}{(M+\gamma s_i)^3(M+\gamma s_j)^3}. \tag{48}$$

Next, we derive $\frac{\partial h_1}{\partial \bar{\gamma}}$. Let

$$D(\bar{\gamma}) = \sum_{i=1}^{n} \frac{\lambda_i a_i^2}{(1+\bar{\gamma}\lambda_i)^2}, \quad E(\bar{\gamma}) = \sum_{i=1}^{n} \frac{a_i^2}{(1+\bar{\gamma}\lambda_i)^2}.$$

We have

$$D'(\bar{\gamma}) = -2\sum_{i=1}^{n} \frac{\lambda_i^2 a_i^2}{(1+\bar{\gamma}\lambda_i)^3}, \quad E'(\bar{\gamma}) = -2\sum_{i=1}^{n} \frac{\lambda_i a_i^2}{(1+\bar{\gamma}\lambda_i)^3}.$$

The derivative of h_1 with respect to $\bar{\gamma}$ is given by

$$\frac{\partial h_1}{\partial \bar{\gamma}} = -\left(\frac{1}{E(\bar{\gamma})}\right)^2 \left(E(\bar{\gamma})D'(\bar{\gamma}) - D(\bar{\gamma})E'(\bar{\gamma})\right). \tag{49}$$

Substituting into (49), we get

$$\left(\frac{1}{E(\bar{\gamma})}\right)^2 = \left(\sum_{i=1}^{n} \frac{a_i^2}{(1+\bar{\gamma}\lambda_i)^2}\right)^{-2}, \tag{50a}$$

and

$$\begin{aligned} E(\bar{\gamma})D'(\bar{\gamma}) - D(\bar{\gamma})E'(\bar{\gamma}) &= 2\sum_{i=1}^{n} \frac{\lambda_i^2 a_i^2}{(1+\bar{\gamma}\lambda_i)^3} \sum_{i=1}^{n} \frac{a_i^2}{(1+\bar{\gamma}\lambda_i)^2} \\ &\quad - 2\sum_{i=1}^{n} \frac{\lambda_i a_i^2}{(1+\bar{\gamma}\lambda_i)^2} \sum_{i=1}^{n} \frac{a_i^2 \lambda_i}{(1+\bar{\gamma}\lambda_i)^3} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\bar{T}_{ij}}{(1+\bar{\gamma}\lambda_i)^3(1+\bar{\gamma}\lambda_j)^3} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{a_i^2 a_j^2 \left(\lambda_i^2 + \lambda_j^2 - 2\lambda_i\lambda_j\right)}{(1+\bar{\gamma}\lambda_i)^3(1+\bar{\gamma}\lambda_j)^3}. \end{aligned} \tag{50b}$$

Here,

$$\begin{aligned} \bar{T}_{ij} &= \lambda_i^2 a_i^2 a_j^2 (1+\bar{\gamma}\lambda_j) + a_i^2(1+\bar{\gamma}\lambda_i)\lambda_j^2 a_j^2 \\ &\quad - \lambda_i a_i^2 (1+\bar{\gamma}\lambda_i) a_j^2 \lambda_j - a_i^2 \lambda_i \lambda_j a_j^2 (1+\bar{\gamma}\lambda_j) \\ &= a_i^2 a_j^2 \left(\lambda_i^2(1+\bar{\gamma}\lambda_j) + (1+\bar{\gamma}\lambda_i)\lambda_j^2 - \lambda_i(1+\bar{\gamma}\lambda_i)\lambda_j - \lambda_i\lambda_j(1+\bar{\gamma}\lambda_j)\right). \end{aligned} \tag{50c}$$

Hence, substituting (50a) and (50b) into (49) gives

$$\frac{\partial h_1}{\partial \bar{\gamma}} = -\left(\sum_{i=1}^{n} \frac{a_i^2}{(1+\bar{\gamma}\lambda_i)^2}\right)^{-2} \sum_{i,j=1}^{n} \frac{a_i^2 a_j^2 (\lambda_i - \lambda_j)^2}{(1+\bar{\gamma}\lambda_i)^3(1+\bar{\gamma}\lambda_j)^3}. \tag{51}$$

Now, for the combined derivative, we have

$$\begin{aligned} \left|\frac{\partial h_2}{\partial \gamma} \cdot \frac{\partial h_1}{\partial \bar{\gamma}}\right| &\leq \left(\sum_{i=1}^{d} \frac{s_i^2}{(M+\gamma s_i)^2}\right)^{-2} \sum_{i,j=1}^{d} \frac{s_i^2 s_j^2 (s_i - s_j)^2}{(M+\gamma s_i)^3 \left(M+\gamma s_j\right)^3} \\ &\quad \cdot \left(\sum_{i=1}^{n} \frac{a_i^2}{(1+\bar{\gamma}\lambda_i)^2}\right)^{-2} \sum_{i,j=1}^{n} \frac{a_i^2 a_j^2 (\lambda_i - \lambda_j)^2}{(1+\bar{\gamma}\lambda_i)^3(1+\bar{\gamma}\lambda_j)^3}. \end{aligned}$$

Note that $M + \gamma s_i$ and $1 + \bar{\gamma}\lambda_j$ are nonnegative for all i, j. Hence,

$$\left|\frac{\partial h_2}{\partial \gamma} \cdot \frac{\partial h_1}{\partial \bar{\gamma}}\right| \leq \left(\sum_{i=1}^{d} \frac{s_i^2 \left(M + \gamma s_i\right)}{\left(M + \gamma s_i\right)^3}\right)^{-2} \cdot \sum_{i,j=1}^{d} \frac{s_i^2 s_j^2 \cdot \Delta_1 \cdot \left(M + \gamma s_j\right)\left(M + \gamma s_i\right)}{\left(M + \gamma s_i\right)^3 \left(M + \gamma s_j\right)^3} \cdot \left(\sum_{i=1}^{n} \frac{a_i^2 (1 + \bar{\gamma}\lambda_i)}{(1 + \bar{\gamma}\lambda_i)^3}\right)^{-2} \cdot \sum_{i,j=1}^{d} \frac{a_i^2 a_j^2 \cdot \Delta_2 \cdot (1 + \bar{\gamma}\lambda_i)\left(1 + \bar{\gamma}\lambda_j\right)}{(1 + \bar{\gamma}\lambda_i)^3 \left(1 + \bar{\gamma}\lambda_j\right)^3},$$

where

$$\Delta_1 := \max_{i,j \in \mathcal{S}} \frac{(s_i - s_j)^2}{\left(M + \gamma s_j\right)\left(M + \gamma s_i\right)}, \quad \Delta_2 := \max_{i,j \in \mathcal{S}} \frac{(\lambda_i - \lambda_j)^2}{(1 + \bar{\gamma}\lambda_i)(1 + \bar{\gamma}\lambda_j)}. \tag{52}$$

Here, $\mathcal{S} = \{i \in [n] \mid \lambda_i \neq 0\} \subset [n]$.

Finally, setting $\bar{\gamma} = h_2(\gamma)$, since $\gamma \geq 1$ and by our assumption $\Delta_{\boldsymbol{\Sigma}} \cdot \Delta_{\boldsymbol{R}} < M + s_{\min}$, we obtain

$$|h_1'(h_2(\gamma)) \cdot h_2'(\gamma)| = \left|\frac{\partial h_2}{\partial \gamma} \cdot \frac{\partial h_1}{\partial h_2}\right| \leq \Delta_1 \cdot \Delta_2 \leq \frac{\Delta_{\boldsymbol{\Sigma}}^2 \cdot \Delta_{\boldsymbol{R}}^2}{(M + s_{\min})^2} < 1.$$

where $\Delta_{\boldsymbol{\Sigma}}$ and $\Delta_{\boldsymbol{R}}$ are the spectral gaps of $\boldsymbol{\Sigma}$ and $\boldsymbol{R}$; and $s_{\min}$ is the smallest eigenvalue of $\boldsymbol{\Sigma}$; and $M = \sigma^2 + \sum_{i=1}^{d} s_i$. □

Proof of Theorem 3. It follows from Lemma 1 that there exists $\gamma^\star \geq 1$ such that $h_1(h_2(\gamma^\star)) + 1 = \gamma^\star$. Further, Lemma 2 establishes that the mapping $h_1(h_2(\gamma)) + 1$ is a contraction mapping, since it satisfies $|\partial h_1(h_2(\gamma))/\partial\gamma| < 1$. Therefore, by the Banach fixed-point theorem, there exists a *unique* fixed-point solution, denoted by $\gamma^\star$, satisfying $\gamma^\star = h_1(h_2(\gamma^\star)) + 1$. This completes the proof of statement **T1.**.

Next, we establish **T2.**. First, from Theorem 2, the stationary point $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$, up to rescaling, is defined as

$$\boldsymbol{P}^\star = \boldsymbol{\Sigma}^{-\frac{1}{2}} \left(\frac{\gamma^\star}{M} \cdot \boldsymbol{\Sigma} + \boldsymbol{I}\right)^{-1} \boldsymbol{\Sigma}^{-\frac{1}{2}} \quad \text{and} \quad \boldsymbol{\omega}^\star = \left(h_2(\gamma^\star) \cdot \boldsymbol{R} + \boldsymbol{I}\right)^{-1} \boldsymbol{r},$$

where $\gamma^\star$ is a fixed point of composite function $h_1(h_2(\gamma)) + 1$.

Given the coercive nature of the loss function $\mathcal{L}(\boldsymbol{P}, \boldsymbol{\omega})$, a global minimum exists. Since the global minimum must satisfy first-order optimality conditions, it must lie within the set of stationary points characterized by Theorem 2.

Now, consider arbitrary global minimizers $(\hat{\boldsymbol{P}}, \hat{\boldsymbol{\omega}})$. Define scaling factors $\alpha, \beta > 0$ and write $(\hat{\boldsymbol{P}}, \hat{\boldsymbol{\omega}}) = (\alpha \boldsymbol{P}^\star, \beta \boldsymbol{\omega}^\star)$. Substituting this scaling into the loss function gives:

$$\mathcal{L}(\alpha \boldsymbol{P}^\star, \beta \boldsymbol{\omega}^\star) = M - 2\alpha\beta \mathtt{tr}\left(\boldsymbol{\Sigma}^2 \boldsymbol{P}^\star\right) \boldsymbol{\omega}^{\star\top} \boldsymbol{r} + M\alpha^2\beta^2 \|\boldsymbol{\omega}^\star\|^2 \mathtt{tr}\left(\boldsymbol{\Sigma} \boldsymbol{P}^{\star\top} \boldsymbol{\Sigma} \boldsymbol{P}^\star\right) + \alpha^2\beta^2 (\|\boldsymbol{\omega}^\star\|^2 + \boldsymbol{\omega}^{\star\top} \boldsymbol{R} \boldsymbol{\omega}^\star) \mathtt{tr}\left(\boldsymbol{\Sigma}^2 \boldsymbol{P}^{\star\top} \boldsymbol{\Sigma} \boldsymbol{P}^\star\right). \tag{53}$$

Differentiating this expression with respect to α and β and setting the derivatives equal to zero, we obtain the equation

$$-2A + 2\alpha\beta(MCB + (C + D)E) = 0, \tag{54a}$$

where we define

$$A := \mathtt{tr}\left(\boldsymbol{\Sigma}^2 \boldsymbol{P}^\star\right) \boldsymbol{\omega}^{\star\top} \boldsymbol{r}, \quad B := \mathtt{tr}\left(\boldsymbol{\Sigma} \boldsymbol{P}^{\star\top} \boldsymbol{\Sigma} \boldsymbol{P}^\star\right), \quad C := \|\boldsymbol{\omega}^\star\|^2, \quad D := \boldsymbol{\omega}^{\star\top} \boldsymbol{R} \boldsymbol{\omega}^\star, \quad E := \mathtt{tr}\left(\boldsymbol{\Sigma}^2 \boldsymbol{P}^{\star\top} \boldsymbol{\Sigma} \boldsymbol{P}^\star\right). \tag{54b}$$

Equation (54) implies a unique constraint on the product $\alpha\beta$, ensuring no independent rescaling beyond a fixed ratio can further minimize the loss. Hence, no distinct minimizers exist other than scaling the original $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$ pair. Thus, the stationary point $(\boldsymbol{P}^\star, \boldsymbol{\omega}^\star)$ from Theorem 2 is indeed a unique minimizer up to rescaling.

□

E.3 Proof of Corollary 1

Proof. Since by assumption $\boldsymbol{\Sigma} = \boldsymbol{I}$, it follows from (14b) that

$$h_2(\gamma^\star) = \left(1 + (d+\sigma^2)\sum_{i=1}^{d} \frac{1}{(d+\sigma^2+\gamma^\star+1)^2}\left(\sum_{i=1}^{d}\frac{1}{(d+\sigma^2+\gamma^\star+1)^2}\right)^{-1}\right)^{-1}$$
$$= \frac{1}{d+\sigma^2+1}.$$

Substituting this into (15) gives

$$\boldsymbol{P}^\star = \boldsymbol{I}, \quad \text{and} \quad \boldsymbol{\omega}^\star = \left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}.$$

Now, recall that the objective function is given by

$$\mathcal{L}(\boldsymbol{\omega}) = M - 2\mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}\right)\boldsymbol{\omega}^\top\boldsymbol{r} + M\|\boldsymbol{\omega}\|_{\ell_2}^2\,\mathtt{tr}\left(\boldsymbol{\Sigma}\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right) + (\|\boldsymbol{\omega}\|_{\ell_2}^2 + \boldsymbol{\omega}^\top\boldsymbol{R}\boldsymbol{\omega})\mathtt{tr}\left(\boldsymbol{\Sigma}^2\boldsymbol{P}^\top\boldsymbol{\Sigma}\boldsymbol{P}\right),$$

and, by assumption, $M = \sigma^2 + d$.

Substituting $\boldsymbol{P}^\star = \boldsymbol{I}$ and $\boldsymbol{\omega}^\star = \left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}$ into the objective (37), and using $\boldsymbol{\Sigma} = \boldsymbol{I}$, we get:

$$\mathcal{L}(\boldsymbol{\omega}^\star) = (\sigma^2+d) - 2\cdot d\cdot\boldsymbol{r}^\top\boldsymbol{\omega}^\star + (\sigma^2+d)\cdot\|\boldsymbol{\omega}^\star\|^2 d + d\left(\|\boldsymbol{\omega}^\star\|^2 + \boldsymbol{\omega}^{\star\top}\boldsymbol{R}\boldsymbol{\omega}^\star\right).$$

The expression simplifies as

$$\mathcal{L}(\boldsymbol{\omega}^\star) = (\sigma^2+d) - 2d\cdot\boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r} + (\sigma^2+d)d\|\boldsymbol{\omega}^\star\|^2 + d\left(\|\boldsymbol{\omega}^\star\|^2 + \boldsymbol{\omega}^{\star\top}\boldsymbol{R}\boldsymbol{\omega}^\star\right).$$

Next, we compute $\|\boldsymbol{\omega}^\star\|^2$ and $\boldsymbol{\omega}^{\star\top}\boldsymbol{R}\boldsymbol{\omega}^\star$. By definition, we have

$$\|\boldsymbol{\omega}^\star\|^2 = \boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-2}\boldsymbol{r},$$

and

$$\boldsymbol{\omega}^{\star\top}\boldsymbol{R}\boldsymbol{\omega}^\star = \boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{R}\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}.$$

Thus,

$$(d+\sigma^2+1)\|\boldsymbol{\omega}^\star\|^2 + \boldsymbol{\omega}^{\star\top}\boldsymbol{R}\boldsymbol{\omega}^\star = \boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}$$
$$\cdot\left((d+\sigma^2+1)\boldsymbol{I} + \boldsymbol{R}\right)\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}$$
$$= \boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}.$$

Substituting this result back into the objective function gives

$$\mathcal{L}(\boldsymbol{\omega}^\star) = (\sigma^2+d) - d\cdot\boldsymbol{r}^\top\left(\boldsymbol{R} + (d+\sigma^2+1)\boldsymbol{I}\right)^{-1}\boldsymbol{r}.$$

□

F Loss landscape of 1-layer GLA

F.1 Proof of Theorem 4

Proof. We first prove that under Assumption 2, $\mathcal{L}^\star_{\texttt{GLA}} = \min_{\boldsymbol{P}\in\mathbb{R}^{d\times d},\boldsymbol{\omega}\in\mathcal{W}} \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P},\boldsymbol{\omega})$ where $\mathcal{W}$ is the search space of weighting vector $\boldsymbol{\omega}\in\mathbb{R}^n$ defined as

$$\mathcal{W} := \left\{ \begin{bmatrix}\omega_1\mathbf{1}_{n_1}^\top & \cdots & \omega_K\mathbf{1}_{n_K}^\top\end{bmatrix}^\top \in \mathbb{R}^n \;\middle|\; 0\le\omega_i\le\omega_j\le 1,\ \forall 1\le i\le j\le K \right\}.$$

Step 1: Define a set $\bar{\mathcal{W}} := \left\{ \begin{bmatrix}\omega_1 & \cdots & \omega_n\end{bmatrix}^\top \in \mathbb{R}^n \;\middle|\; 0\le\omega_i\le\omega_j\le 1,\ \forall 1\le i\le j\le n \right\}$ and we have $\mathcal{W}\in\bar{\mathcal{W}}$. Given scalar gating $\boldsymbol{G}_i = \begin{bmatrix} * & * \\ g_i\mathbf{1}^\top & * \end{bmatrix}$, following (11), the weighting vector returns

$$\boldsymbol{\omega} := \begin{bmatrix} g_{1:n+1} & \cdots & g_{n:n+1}\end{bmatrix}^\top.$$

Since GLA with scalar gating valued in $[0,1]$ following Assumption 2, that is, $g_i\in[0,1]$, we have $g_{i:n+1}\le g_{j:n+1}$ for $1\le i\le j\le n$. Therefore, any weighting vector $\boldsymbol{\omega}$ implemented by GLA gating should be inside $\bar{\mathcal{W}}$.

Step 2: Next, we will show that

$$\boldsymbol{\omega}^\star\in\mathcal{W} \quad \text{where} \quad \boldsymbol{\omega}^\star = \arg\min_{\boldsymbol{P},\boldsymbol{\omega}\in\bar{\mathcal{W}}} \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P},\boldsymbol{\omega}).$$

Define the weighting vector $\boldsymbol{\omega} = [\boldsymbol{\omega}_1^\top \ \cdots \ \boldsymbol{\omega}_K^\top]^\top \in \mathbb{R}^n$ where we have $\boldsymbol{\omega}_k = [\omega_1^{(k)} \ \cdots \ \omega_{n_k}^{(k)}]^\top \in \mathbb{R}^{n_k}$. For any $\boldsymbol{\omega}\notin\mathcal{W}$, there exist (i,j,k) with $i=j-1$ such that $\omega_i^{(k)} < \omega_j^{(k)}$. Given gradient in (41), we have that

$$\nabla_{\omega_i^{(k)}}\mathcal{L} - \nabla_{\omega_j^{(k)}}\mathcal{L} = 2\left(M\texttt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right) + \texttt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)\right)(\omega_i^{(k)} - \omega_j^{(k)}).$$

Here, (42a) gives that (optimal) $\tilde{\boldsymbol{P}} \neq 0$ and therefore, we have that $\left(M\texttt{tr}\left(\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right) + \texttt{tr}\left(\boldsymbol{\Sigma}\tilde{\boldsymbol{P}}^\top\tilde{\boldsymbol{P}}\right)\right) > 0$ and $\nabla_{\omega_i^{(k)}}\mathcal{L} < \nabla_{\omega_j^{(k)}}\mathcal{L}$. Therefore either increasing $\omega_i^{(k)}$ (if $\nabla_{\omega_i^{(k)}}\mathcal{L} < 0$) or decreasing $\omega_j^{(k)}$ (if $\nabla_{\omega_j^{(k)}}\mathcal{L} > 0$) will reduce the loss. This results in showing that the optimal weighting vector $\boldsymbol{\omega}^\star$ satisfies $\omega_i^{(k)} = \omega_j^{(k)}$ for any $i,j\in[n_k]$ and $k\in[K]$. Hence, $\boldsymbol{\omega}^\star\in\mathcal{W}$.

Step 3: Finally, we will show that any $\boldsymbol{\omega}\in\mathcal{W}$ can be obtained via the GLA gating. Let $\boldsymbol{\omega} = [\omega_1\mathbf{1}_{n_1}^\top \ \cdots \ \omega_K\mathbf{1}_{n_K}^\top]^\top$ be any vector in $\mathcal{W}$ and assume that $\omega_K = \alpha < 1$ without loss of generality. Then such sample weighting can be achieved via the gating

$$\begin{bmatrix}\mathbf{1}_{n_1}^\top & \frac{\omega_1}{\omega_{1:K}} & \cdots & \mathbf{1}_{n_k}^\top & \frac{\omega_k}{\omega_{k:K}} & \cdots & \mathbf{1}_{n_K}^\top & \frac{\omega_K}{\omega_{K:K}}\end{bmatrix}^\top.$$

Let $\omega_k' := \frac{\omega_k}{\omega_{k:K}}$ and let $\boldsymbol{w}_g$ be in the form of

$$\boldsymbol{w}_g = \begin{bmatrix}\mathbf{0}_{d+1} \\ \tilde{\boldsymbol{w}}_g\end{bmatrix} \in \mathbb{R}^{d+p+1}.$$

Then, it remains to show that there exists $\tilde{\boldsymbol{w}}_g$ satisfying

$$\phi(\tilde{\boldsymbol{w}}_g^\top\bar{\boldsymbol{c}}_k) = \begin{cases}1, & k=0,\\ \omega_k', & k\in[K].\end{cases}$$

The linear independence assumption in Assumption 2 implies that the problem is feasible, which completes the proof of (22).

Proof of (23): Recap the optimal weighting from (15) which takes the form of

$$\boldsymbol{\omega}^\star = \left(h_2(\gamma^\star)\cdot \boldsymbol{R} + \boldsymbol{I}\right)^{-1}\boldsymbol{r},$$

where $h_2(\gamma^\star) > 0$. Since Assumption 3 holds and $n_1 = n_2 = \cdots = n_K := \bar{n}$, $\boldsymbol{R}$ is block diagonal matrix, where each block is an all-ones matrix. That is

$$\boldsymbol{R} = \begin{bmatrix} \mathbf{1}_{\bar{n}\times\bar{n}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{\bar{n}\times\bar{n}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{\bar{n}\times\bar{n}} \end{bmatrix}.$$

Therefore, we get

$$\left(h_2(\gamma^\star)\cdot \boldsymbol{R} + \boldsymbol{I}\right)^{-1} = \boldsymbol{I} - \frac{h_2(\gamma^\star)}{h_2(\gamma^\star)\cdot\bar{n}+1}\boldsymbol{R}.$$

Since $\boldsymbol{R}\boldsymbol{r} = \bar{n}\boldsymbol{r}$, then

$$\boldsymbol{\omega}^\star = \frac{1}{h_2(\gamma^\star)\cdot\bar{n}+1}\boldsymbol{r}.$$

Therefore, the optimal weighting (up to a scalar) is inside the set $\mathcal{W}$. Combining it with (22) completes the proof.

□

F.2 Proof of Theorem 5

Proof. Following the similar proof of Theorem 4, and letting $\tilde{\mathcal{W}} := \left\{\left[\omega_1\mathbf{1}_{n_1}^\top \ \cdots \ \omega_K\mathbf{1}_{n_K}^\top\right]^\top \in \mathbb{R}^n\right\}$, we obtain

$$\min_{\boldsymbol{P}\in\mathbb{R}^{d\times d},\boldsymbol{\omega}\in\tilde{\mathcal{W}}} \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P},\boldsymbol{\omega}) = \min_{\boldsymbol{P}\in\mathbb{R}^{d\times d},\boldsymbol{\omega}\in\mathbb{R}^n} \mathcal{L}_{\texttt{WPGD}}(\boldsymbol{P},\boldsymbol{\omega}).$$

Therefore, it remains to show that any $\boldsymbol{\omega}\in\tilde{\mathcal{W}}$ can be implemented via some gating function. Let $\boldsymbol{\omega} = [\omega_1\mathbf{1}_{n_1}^\top \ \cdots \ \omega_K\mathbf{1}_{n_K}^\top]^\top$ be arbitrary weighting in $\tilde{\mathcal{W}}$. Theorem 4 has shown that if $\omega_1 \le \omega_2 \le \cdots \le \omega_K$, GLA with scalar function can implement such increasing weighting.

Now, inspired from Appendix D, **Construction 2** and (29) that all dimensions in the output implement individual WPGD, the weighting can be a composition of up to $d+1$ different weights ($\boldsymbol{u}$ is $(d+1)$-dimensional). Therefore, for any $\omega_1,\cdots,\omega_K \in \mathbb{R}$, we can get K separate weighting:

$$\begin{aligned}
\boldsymbol{\omega}_1 &= \omega_1[\mathbf{1}_{n_1}^\top \ \cdots \ \mathbf{1}_{n_K}^\top]^\top, \\
\boldsymbol{\omega}_2 &= (\omega_2-\omega_1)[\mathbf{0}_{n_1}^\top \ \mathbf{1}_{n_2}^\top \ \cdots \ \mathbf{1}_{n_K}^\top]^\top, \\
\boldsymbol{\omega}_3 &= (\omega_3-\omega_2)[\mathbf{0}_{n_1}^\top \ \mathbf{0}_{n_2}^\top \ \mathbf{1}_{n_3}^\top \ \cdots \ \mathbf{1}_{n_K}^\top]^\top, \\
&\vdots \\
\boldsymbol{\omega}_K &= (\omega_K-\omega_{K-1})[\mathbf{0}_{n_1}^\top \ \mathbf{0}_{n_2}^\top \ \mathbf{0}_{n_3}^\top \ \cdots \ \mathbf{0}_{n_{K-1}}^\top \ \mathbf{1}_{n_K}^\top]^\top.
\end{aligned}$$

Recap from Appendix D and (29), and consider the construction $\tilde{\boldsymbol{W}}_v = [\mathbf{0}_{(d+p+1)\times d} \ \boldsymbol{u} \ \mathbf{0}_{(d+p+1)\times p}]^\top$. Assumption 2 implies that $K \le p < d+p+1$.

From (28) and (29), let i'th dimension implements the weighting $\boldsymbol{\omega}_i$ for $i\in[K]$. Specifically, let i'th row of each gating matrix $\boldsymbol{G}_i$ implement weighting $[\mathbf{0}_{n_1}^\top \ \cdots \ \mathbf{0}_{n_{i-1}}^\top \ \mathbf{1}_{n_i}^\top \ \cdots \ \mathbf{1}_{n_k}^\top]$ (which is feasible due to Theorem 4) and set $u_i h_i = \omega_i - \omega_{i-1}$ with $\omega_0 = 0$. Then the composed weighting following (29) returns $\boldsymbol{\omega} = \sum_{k=1}^K \boldsymbol{\omega}_k$, which completes the proof. □

F.3 Proof of Corollary 2

Proof. Recap from (42a) that given $\boldsymbol{\Sigma} = \boldsymbol{I}$ and $\boldsymbol{\omega} = \mathbf{1}$,

$$\begin{aligned}\boldsymbol{P}^\star &= \Big((d+\sigma^2)n\boldsymbol{I} + (n + \mathbf{1}^\top \boldsymbol{R}\mathbf{1})\boldsymbol{I}\Big)^{-1}\mathbf{1}^\top \boldsymbol{r} \\ &= \frac{\mathbf{1}^\top \boldsymbol{r}}{n(d+\sigma^2+1) + \mathbf{1}^\top \boldsymbol{R}\mathbf{1}}\boldsymbol{I} := c\boldsymbol{I}.\end{aligned}$$

Then taking it back to the loss function (c.f. (37)) obtains

$$\begin{aligned}\mathcal{L}(\boldsymbol{P}^\star, \boldsymbol{\omega} = 1) &= d + \sigma^2 - 2cd\mathbf{1}^\top \boldsymbol{r} + (d+\sigma^2)c^2 nd + (n + \mathbf{1}^\top \boldsymbol{R}\mathbf{1})c^2 d \\ &= d + \sigma^2 - cd\mathbf{1}^\top \boldsymbol{r}.\end{aligned}$$

It completes the proof. □