CONVEXIFYING TRANSFORMERS: IMPROVING OPTIMIZATION AND UNDERSTANDING OF TRANSFORMER NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the fundamental mechanism behind the success of transformer networks is still an open problem in the deep learning literature. Although their remarkable performance has been mostly attributed to the self-attention mechanism, the literature still lacks a solid analysis of these networks and interpretation of the functions learned by them. To this end, we study the training problem of attention/transformer networks and introduce a novel convex analytic approach to improve the understanding and optimization of these networks. Particularly, we first introduce a convex alternative to the self-attention mechanism and reformulate the regularized training problem of attention/transformer networks. Then, we cast the reformulation as a convex optimization problem that is interpretable and easier to optimize. Moreover, as a byproduct of our convex analysis, we reveal an implicit regularization mechanism, which promotes sparsity across tokens. Therefore, we not only improve the optimization of attention/transformer networks but also provide a solid theoretical understanding of the functions learned by them. We also demonstrate the effectiveness of our theory through several numerical experiments.

1 INTRODUCTION

Transformer networks proposed by Vaswani et al. (2017) have become a dominant architecture in various tasks, especially Natural Language Processing (NLP) (Devlin et al., 2018; Radford et al., 2019), due to their extraordinary generalization properties and high capacity to learn from vast amount of data. Although there exists substantial empirical evidence on the effectiveness of transformer networks, revealing the underlying theoretical reasons behind their success is still an open research problem due to their highly nonlinear and nonconvex structure.

A significant body of research focused on analyzing certain components of transformer networks via empirical studies. As an example, Liu et al. (2021a); Vashishth et al. (2019); Dong et al. (2021); Voita et al. (2019); Takase et al. (2022); Liu et al. (2021a) studied the impact of the attention mechanism on transformer networks. Although these studies agreed that attention is an essential component of transformers, they also raised several issues regarding interpretability and optimization. Particularly, Voita et al. (2019) demonstrated that most attention heads can be removed without affecting the performance of the network, which is an indicator of large amount of redundancy in the network. Vashishth et al. (2019) provided a set of empirical evidence showing that attention might not be needed for some NLP tasks. Additionally, Dong et al. (2021) revealed that although attention is at the heart of transformer networks, training an attention network in the absence of Fully Connected Network (FCN) layers and skip connections is extremely challenging since the network output degenerates quickly without them. Similarly, Takase et al. (2022) discussed the importance of layer normalization and skip connections for transformer networks so that even changing the position of these might considerably impact the performance of a transformer network. However, a solid theoretical analysis of the underlying factors behind these issues is sill lacking, likely due to the highly complex and nonconvex structure of transformer networks.

A series of papers also focused on designing new alternatives to the self-attention mechanism which perform similarly and might provide further interpretations towards the overall model. One set of

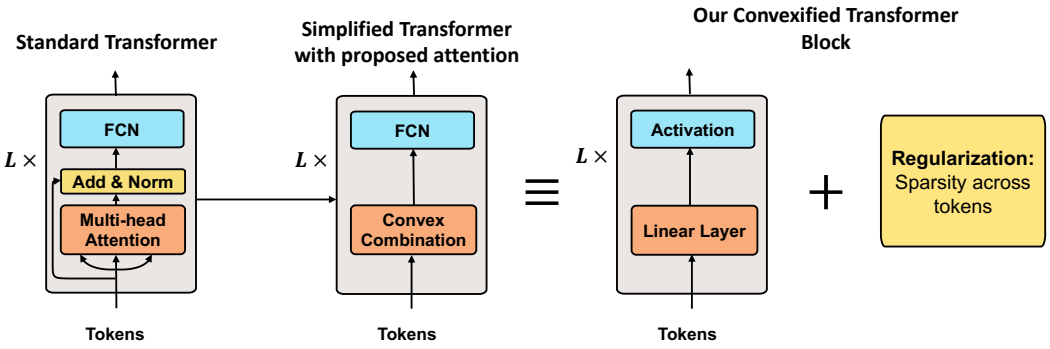


Figure 1: Summary of our main findings: We first propose an alternative to attention, i.e., taking the convex combinations of tokens, and then convexifying whole transformer block (attention + Fully Connected Network (FCN)) with this new attention mechanism. The equivalent convex formulation also reveals a sparsity-inducing regularization across tokens as detailed in Theorem 1, 2, and 3.

work utilizes multi-layer perceptron based architectures, Tolstikhin et al. (2021); Tatsunami & Taki (2021); Touvron et al. (2021); Liu et al. (2021b); Yu et al. (2021), while another set of of papers proposes Fourier based models Lee-Thorp et al. (2021); Rao et al. (2021); Li et al. (2020); Guibas et al. (2021). Others also proposed replacing the self-attention mechanism with matrix decomposition Geng et al. (2021). Although these works successfully applied to certain applications, they lack any solid theoretical analysis and understanding from an optimization perspective.

Recently, another line of research has focused on understanding structures and patterns emerge throughout the training of transformer networks (Power et al., 2022; Thilak et al., 2022; Barak et al., 2022). In particular, the grokking phenomenon was first observed by Power et al. (2022) on specific algorithmic tasks, such as modular division operations. Specifically, grokking refers to a sudden transition of validation or test accuracy to perfect generalization and this generalization happens well past the point of perfect training accuracy. This interesting behavior contradicts the common practice of early stopping in the training of deep learning models and definitely requires further understanding as to why this phenomenon emerges.

In order to remedy the issues associated with the standard transformer networks, in this paper, we develop a convex optimization perspective to train, analyze and understand transformer networks. Particularly, we first propose a convex alternative to the self-attention mechanism and then develop our convex analytic framework on the resulting model as detailed in Figure 1.

1.1 Contributions

Our contributions can be summarized as follows:

- We propose an alternative formulation to the standard self-attention mechanism and study the regularized training problem of attention/transformer networks with it.
- We convexify the regularized training problem of attention/transformer networks with the proposed attention layer as shown in Figure 1 and therefore enable finding a globally optimal solution without requiring any nonconvex optimization heuristic, e.g., layer normalization and skip connections.
- We also apply our convex analytic framework to various architectures, e.g., networks with or without an FCN layer. Thus, we are able to explain the impact of each component on the models learned throughout training.
- We reveal an implicit regularization mechanism induced by our attention mechanism. We further characterize this regularization as a sparsity-inducing factor across tokens.
- We demonstrate the effectiveness of our convex reformulation via various experimental results. We also show that our reformulation significantly mitigates the grokking phenomenon studied in recent papers (Power et al., 2022; Thilak et al., 2022).

1.2 NOTATIONS

We use lowercase and uppercase bold letters to denote vectors and matrices, respectively. We denote a certain column/element of a vector or matrix using subscripts. As an example, w_{jk} denotes the jk^{th} entry of the matrix $\mathbf{W}$. We use $\mathbf{I}_k$ to denote the identity matrix of size $k \times k$ and $\mathbf{0}$ (or $\mathbf{1}$) to denote a vector/matrix of zeros (or ones) with appropriate sizes. We also use $[n]$ for the set of integers ranging from 1 to n. We represent the Euclidean and Frobenius norms as $\|\cdot\|_2$ and $\|\cdot\|_F$, respectively. We also use $\mathbb{1}[x \geq 0]$ to denote the 0-1 valued indicator function. We provide more notations we use throughout the paper in Table 1.

Table 1: Notations.

Notation	Description
N	# of sentences/samples
n	# of tokens
d	embedding dimension
h	# of heads
c	# of outputs

2 TRANSFORMER NETWORKS

Given a data sample (or sentence) $\mathbf{X} \in \mathbb{R}^{h \times d}$ as a sequence of h tokens with the embedding dimension d, we define the key, query, and value matrices as

$$\begin{aligned} \mathbf{Q} &= \mathbf{X}\mathbf{W}_q, \quad \mathbf{W}_q \in \mathbb{R}^{d \times d} \\ \mathbf{K} &= \mathbf{X}\mathbf{W}_k, \quad \mathbf{W}_k \in \mathbb{R}^{d \times d}, \\ \mathbf{V} &= \mathbf{X}\mathbf{W}_v, \quad \mathbf{W}_v \in \mathbb{R}^{d \times d} \end{aligned}$$

which are the main components of the self-attention mechanism. Then, a single transformer block, which is basically a stack of self attention, residual connection, layer normalization, and point-wise feedforward connections, can be formulated as follows

$$\begin{aligned} \mathbf{A}_{s,j} &= \text{softmax}\left(\mathbf{Q}\mathbf{K}^\top\right)\mathbf{V} \\ \mathbf{A}_o &= \mathbf{A}_s\mathbf{W}_o, \quad \mathbf{W}_o \in \mathbb{R}^{d \times d} \\ \mathbf{X}_A &= \text{LayerNorm}\left(\mathbf{A}_o\right) + \mathbf{X} \\ \mathbf{X}_B &= \sigma\left(\mathbf{X}_A\mathbf{W}_1\right)\mathbf{W}_2 \end{aligned},$$

where $\sigma(\cdot)$ denotes the activation function for the FCN layer. Although skip connections, layer normalization and FCN also play a crucial role in a transformer block, the success of these networks has been mostly attributed to the self-attention part, denoted as $\mathbf{A}_o$ (Vaswani et al., 2017). Therefore, in the following section, we first study the training problem of a simplified transformer network, for which the network output is directly $\mathbf{A}_o$. We then extend our derivations to a transformer network with FCN layers.

3 ATTENTION-ONLY NETWORKS

We first consider a simplified transformer network only with a self attention layer that maps input sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$ to the output sequence $\hat{\mathbf{Y}} \in \mathbb{R}^{n \times c}$ with c outputs as follows

$$\hat{\mathbf{Y}} = \text{softmax}\left(\mathbf{X}\mathbf{W}_q\mathbf{W}_k^\top\mathbf{X}^\top\right)\mathbf{X}\mathbf{W}_v\mathbf{W}_o. \tag{1}$$

We also call the model (1) as an attention-only network. This is a meaningful model and has been applied to various tasks, including machine translation, language modeling, image captioning, and object recognition (Vashishth et al., 2019).

We next consider a standard regression framework with an arbitrary convex loss function. Given a training set $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^N$, where $\mathbf{X}_i \in \mathbb{R}^{n \times d}$ and $\mathbf{Y}_i \in \mathbb{R}^{n \times c}$ denote the input sequence and the labels/target outputs, respectively, the weight decay regularized training problem for the attention-only network in (1) is as follows

$$\min_{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o} \sum_{i=1}^{N} \mathcal{L}\left(\text{softmax}\left(\mathbf{X}_i\mathbf{W}_q\mathbf{W}_k^\top\mathbf{X}_i^\top\right)\mathbf{X}\mathbf{W}_v\mathbf{W}_o, \mathbf{Y}_i\right) + \frac{\beta}{2}\sum_{\#\in\{q,k,v,o\}} \|\mathbf{W}_\#\|_F^2, \tag{2}$$

where $\mathcal{L}(\cdot)$ is an arbitrary convex loss function, including squared loss and cross entropy, and $\beta > 0$ is the regularization coefficient.

Although the attention-only model in (1) is quite powerful across varios NLP tasks, e.g., natural language inference, neural machine translation, and text classification (Vashishth et al., 2019), the corresponding training problem in (2) is an extremely challenging optimization task and requires various nonconvex optimization heuristics (Dong et al., 2021) to be adequately trained. To remedy these issues, in the following sections, we first reformulate the training problem by replacing the attention part with an alternative convex layer and then cast the reformulated training problem as an interpretable convex optimization problem that enables the globally optimizing the network parameters.

3.1 Convex attention layer

We first note that since the $\mathrm{softmax}(\cdot)$ operation is highly nonlinear and nonconvex, the training problem in (2) is a challenging nonconvex optimization problem. Therefore, one may not adequately train attention networks and obtain trivial models at the end of training. For example, Dong et al. (2021) shows that attention networks are likely to degenerate throughout the training and the output converges to a rank-1 matrix. Thus, they fail to learn the underlying tasks.

To avoid the issues associated with the nonconvex formulation in (1), we first replace the softmax operation with a simpler yet effective alternative. Particularly, since softmax converts the rows of its input matrix to a probability distribution, it can be relaxed as a linear operation with unit simplex constraints as follows

$$\text{for any } \mathbf{U} \in \mathbb{R}^{n \times n}, \quad \exists \mathbf{W} \in \Delta \text{ s.t. } \mathrm{softmax}(\mathbf{U})\,\mathbf{X} = \mathbf{W}\mathbf{X},$$

where $\Delta := \{\mathbf{W} \in \mathbb{R}^{n \times n} : \mathbf{w}_i \geq \mathbf{0}, \mathbf{1}^\top \mathbf{w}_i = 1, \forall i \in [n]\}$ denotes a convex set of constraints, also termed as unit simplex constraints. Thus, we simplified and convexified the attention mechanism without disturbing its structure. Based on this observation, (2) can be reformulated as follows

$$\min_{\substack{\mathbf{W}_1 \in \Delta \\ \mathbf{W}_2 \in \mathbb{R}^{d \times d}, \mathbf{W}_3 \in \mathbb{R}^{d \times c}}} \sum_{i=1}^{N} \mathcal{L}\left(\mathbf{W}_1 \mathbf{X}_i \mathbf{W}_2 \mathbf{W}_3, \mathbf{Y}_i\right) + \frac{\beta}{2}\left(\|\mathbf{W}_2\|_F^2 + \|\mathbf{W}_3\|_F^2\right). \tag{3}$$

Note that the model above utilizes a single head attention model and, therefore, may not be practically relevant due to its insufficient expressive power. Thus, we introduce the concept of head to the problem in (3) as follows

$$\min_{\substack{\mathbf{W}_{1j} \in \Delta \\ \mathbf{W}_{2j} \in \mathbb{R}^{d \times d}, \mathbf{W}_{3j} \in \mathbb{R}^{d \times c}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{W}_{1j} \mathbf{X}_i \mathbf{W}_{2j} \mathbf{W}_{3j}, \mathbf{Y}_i\right) + \frac{\beta}{2}\left(\sum_{j=1}^{h} \|\mathbf{W}_{2j}\|_F^2 + \|\mathbf{W}_{3j}\|_F^2\right). \tag{4}$$

Now, we are ready to apply the convex analytic tools to (4) as detailed in the next section.

3.2 Convex optimization for attention-only networks

As a warm-up, let us consider the scalar output prediction problem where the targets are one-dimensional, i.e., $y_i \in \mathbb{R}$. Then, (4) reduces to the following optimization problem

$$\min_{\substack{\mathbf{w}_{1j} \in \Delta \\ \mathbf{w}_{2j} \in \mathbb{R}^d, w_{3j} \in \mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, y_i\right) + \frac{\beta}{2} \sum_{j=1}^{h} \left(\|\mathbf{w}_{2j}\|_2^2 + (w_{3j})^2\right). \tag{5}$$

Next, we first apply a rescaling between the parameters $\mathbf{w}_{2j}$ and w_{3j} such that (5) can be described as an ℓ_1 regularized optimization problem.

Lemma 1. *The problem in (5) is equivalent to the following ℓ_1 regularized training problem*

$$\min_{\substack{\mathbf{w}_{1j} \in \Delta \\ \|\mathbf{w}_{2j}\|_2 \leq 1, w_{3j} \in \mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, y_i\right) + \beta \|\mathbf{w}_3\|_1. \tag{6}$$

Table 2: Number of parameters and FLOPs for the convex and nonconvex models. Here, we use the following notations: n: # of tokens, d: embedding dimension, h: # of heads, and c: # of outputs.

	Nonconvex				**Convex**	
	Standard		**Alternative (Ours)**			
	# of params	**FLOPs**	**# of params**	**FLOPs**	**# of params**	**FLOPs**
Scalar output	$h(3d^2+d)$	$\mathcal{O}(n^2dh)$	$h(n+d+1)$	$\mathcal{O}(nd)$	nd	$\mathcal{O}(nd)$
Multi output	$h(3d^2+dc)$	$\mathcal{O}(n^2dh+ndhc)$	$h(n+d+c)$	$\mathcal{O}(nd+c)$	ndc	$\mathcal{O}(ndc)$
Multi output with FCN	$h(3d^2+dc)$	$\mathcal{O}(n^2dh+ndhc)$	$h(n+d+c)$	$\mathcal{O}(nd+c)$	$ndch$	$\mathcal{O}(ndch)$

Based on the equivalent formulation in Lemma 1, the next theorem introduces a convex optimization problem that is equivalent to (5).

Theorem 1. *The nonconvex optimization problem (5) can be equivalently cast as the following convex optimization problem*

$$\min_{\mathbf{Z}\in\mathbb{R}^{n\times d}} \frac{1}{2}\sum_{i=1}^{N}\mathcal{L}\left(\text{trace}\left(\mathbf{Z}^\top\mathbf{X}_i\right),y_i\right)+\beta\sum_{k=1}^{n}\left\|\mathbf{z}_k\right\|_2. \tag{7}$$

Note that the equivalent convex model in (7) requires a single parameter matrix $\mathbf{Z}\in\mathbb{R}^{n\times d}$, where each row is the attentions scores of the corresponding token. We also remark that the regularization in (7), i.e., the sum of ℓ_2 norms of the rows of the parameter matrix $\mathbf{Z}$, is a specific type of regularization, also known as group ℓ_1 or Lasso, introduced by (Bakin et al., 1999) and shown to promote group sparsity across parameters (Yuan & Lin, 2006). In our case, the group sparsity is across the token index k. Therefore, one can interpret the model in (7) as a sparse linear model, where the sparsity is across tokens. In other words, (7) can be explained as a model that tries to use as few tokens as possible to fit the training labels $\{y_i\}_{i=1}^N$.

Unlike the nonnegative attention scores in (5), denoted as $\mathbf{w}_{1j}\in\Delta$, the convex parameters $\mathbf{Z}\in\mathbb{R}^{n\times d}$ do not require any constraints. Therefore, one can directly apply standard optimization algorithms, such as SGD and Adam, to train the convex optimization problem (7). Moreover, an optimal set of parameters for (5) can be recovered from a solution to (7) as proven in the following result.

Proposition 1. *After solving the convex optimization problem in (7), one can recover an optimal solution to the original nonconvex optimization problem in (5), denoted as $\{\mathbf{w}_{1j}^*,\mathbf{w}_{2j}^*,w_{3j}^*\}_{j=1}^h$, as follows*

$$\mathbf{w}_{1j}^*=\mathbf{e}_j,\ \mathbf{w}_{2j}^*=\frac{\mathbf{z}_j}{\sqrt{\left\|\mathbf{z}_j\right\|_2}},\ w_{3j}^*=\sqrt{\left\|\mathbf{z}_j\right\|_2},\forall j\in[h],$$

where $\mathbf{e}_j\in\mathbb{R}^n$ is the jth ordinary basis vector, $\mathbf{z}_j\in\mathbb{R}^d$ is the jth row of $\mathbf{Z}$, and we assume that there are h nonzero rows out of n rows of $\mathbf{Z}$ due to the sparsity-inducing regularization in (7).

Proposition 1 proves that there is a one-to-one mapping between the parameters of the nonconvex formulation in (5) and the convex formulation in (7). Therefore, there is no need to solve the challenging nonconvex optimization problem (5) which also requires several optimization heuristics to be adequately trained. Instead, one can solve the convex problem (7) and then use the mapping in Proposition 1 to obtain an optimal solution to (5).

3.3 Extension to multidimensional outputs

In the previous section, we considered a setting with scalar target variables, i.e., $y_i\in\mathbb{R}$. However, for some problems, e.g., multiclass classification, target variables can be multidimensional. Therefore, we now extend the analysis to the problems with multiple/vector outputs. Particularly, we consider the following optimization problem

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta\\ \mathbf{w}_{2j}\in\mathbb{R}^d,\mathbf{w}_{3j}\in\mathbb{R}^c}} \sum_{i=1}^{N}\mathcal{L}\left(\sum_{j=1}^{h}\mathbf{w}_{1j}^\top\mathbf{X}_i\mathbf{w}_{2j}\mathbf{w}_{3j},\mathbf{y}_i\right)+\frac{\beta}{2}\sum_{j=1}^{h}\left(\left\|\mathbf{w}_{2j}\right\|_2^2+\left\|\mathbf{w}_{3j}\right\|_1^2\right), \tag{8}$$

where $\mathbf{y}_i \in \mathbb{R}^c$ and c denotes the number of outputs/classes. Note that here we put ℓ_1^2-norm on $\mathbf{w}_{3j}$ to enable our convex arguments but this does not impact performance of the network in practice. Then, following the same derivations yields the convex program in the next result.

Theorem 2. *The nonconvex optimization problem* (8) *is equivalent to the following convex optimization problem*

$$\min_{\mathbf{Z}_l \in \mathbb{R}^{n \times d}} \sum_{i=1}^{N} \sum_{l=1}^{c} \mathcal{L}\left(\text{trace}\left(\mathbf{Z}_l^\top \mathbf{X}_i\right), y_{il}\right) + \beta \sum_{l=1}^{c} \sum_{k=1}^{n} \left\|\mathbf{z}_{lk}\right\|_2 . \tag{9}$$

Theorem 2 basically shows that the equivalent convex model becomes separable over the output index l, i.e., instead of a single parameter matrix in (7), here we have c parameter matrices due to having c outputs in the nonconvex model (8) (see Table 2 for details). This also illustrates that the number of outputs in the network directly controls the overparameterization level of the equivalent convex formulation.

3.4 Attention networks with FCN layers

Although the model in (4) exhibits interesting properties in various applications (Dong et al., 2021), it is basically a linear function of the token matrix $\mathbf{X}$. Therefore, it is likely to suffer from inadequate performance especially for some challenging problems in NLP. A series of papers (Dong et al., 2021; Geva et al., 2021; Meng et al., 2022; Geva et al., 2022b;a) also confirmed the importance of FCNs via extensive empirical evidence. Therefore, in this section, we include an FCN layer to our attention only model in (4) and derive an equivalent convex formulation for this new model.

Here, we consider the following optimization problem

$$\min_{\substack{\mathbf{w}_{1j} \in \Delta \\ \mathbf{w}_{2j}, \mathbf{w}_{3j} \in \mathbb{R}^c}} \sum_{i=1}^{N} \mathcal{L}\left(\sigma\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right) \mathbf{w}_{3j}, \mathbf{y}_i\right) + \frac{\beta}{2}\left(\sum_{j=1}^{h} \left\|\mathbf{w}_{2j}\right\|_2^2 + \left\|\mathbf{w}_{3j}\right\|_1^2\right), \tag{10}$$

where $\sigma\left(\cdot\right)$ is the activation function.

Theorem 3. *The nonconvex optimization problem* (10) *with the gated ReLU activation is equivalent the following convex optimization problem*

$$\min_{\mathbf{Z}_{jl} \in \mathbb{R}^{n \times d}} \sum_{i=1}^{N} \sum_{l=1}^{c} \mathcal{L}\left(\sum_{j=1}^{h} \mathbb{1}_{ij} \text{trace}\left(\mathbf{Z}_{jl}^\top \mathbf{X}_i\right), y_{il}\right) + \beta \sum_{l=1}^{c} \sum_{j=1}^{h} \sum_{k=1}^{n} \left\|\mathbf{z}_{jlk}\right\|_2 , \tag{11}$$

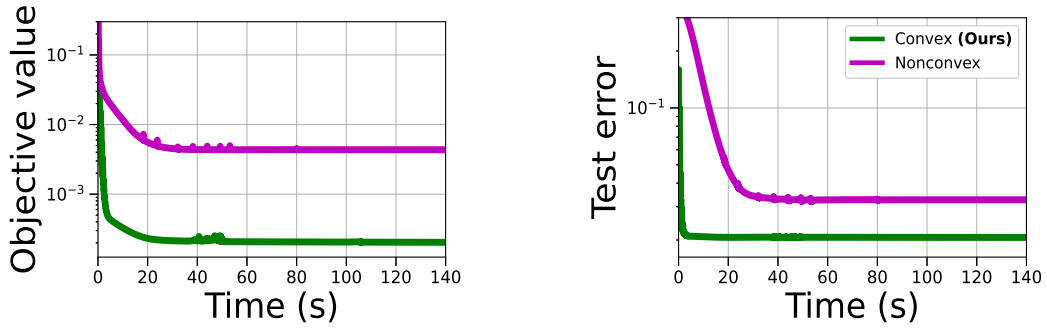
where $\mathbb{1}_{ij} := \mathbb{1}\left\{\mathbf{u}_{1j}^\top \mathbf{X}_i \mathbf{u}_{2j} \geq 0\right\}$ *denotes the indicator function for gated ReLU activation and here* $\left\{\mathbf{u}_{1j}, \mathbf{u}_{2j}\right\}_{j=1}^{h}$ *are fixed vectors that can be randomly selected.*

Theorem 3 implies that introducing the activation function results in further increase in the overparameterization level of the equivalent convex formulation. Precisely, (11) has h times more parameter than (9) as shown in Table 2.

4 Numerical experiments

In this section, we present experimental results corroborating our theory in the previous sections.

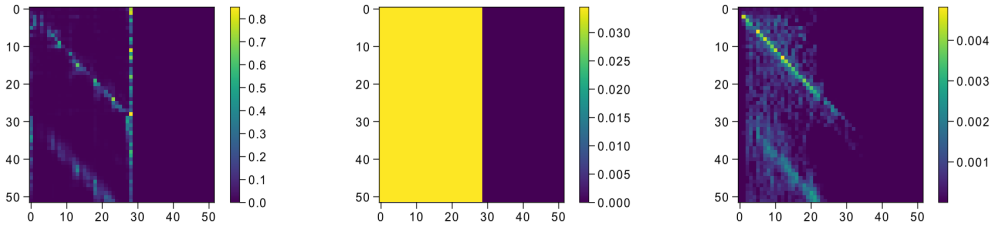
Student-teacher setting with BERT: We first consider a student-teacher setting with the pretrained BERT model in the Hugging Face repository, i.e., `bert-base-uncased`. Particularly, we feed the samples from the mrpc subset of the glue dataset (Warstadt et al., 2018; Wang et al., 2019) through the pretreained BERT model and save the input and output activations in a certain layer. Then, we train the attention-only models, i.e., nonconvex (8) and convex (9), from scratch using these pre and post activations as our training dataset. All the experiments throughout this section are performed using a single GPU on Google Colab. We also use the same regularization coefficient β and optimizer, i.e., Adam, and tune the learning rate and regularization coefficient by performing a grid search on a validation dataset for both algorithms. However, notice that we do not use any nonconvex optimization heuristics, e.g., layer normalization and skip connections, for the convex model



(a) Objective value

(b) Test error

Figure 2: Comparison of the convex and nonconvex models on the dataset extracted from pretrained BERT architecture in a student-teach setting. Here, our convex training approach achieves significantly lower objective value and test error than the original nonconvex training.



(a) Ground truth

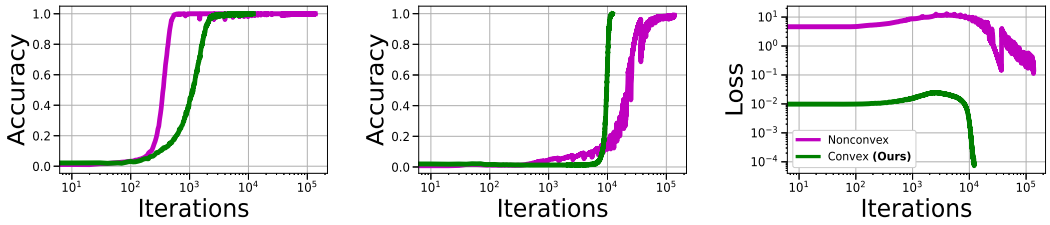
(b) Nonconvex

(c) Convex

Figure 3: Attention maps obtained by our convex and standard nonconvex training approaches as well as the ground truth attention map in the BERT model. Here, we show that the nonconvex training fails to learn the underlying patterns and simply achieves a uniform attention map. However, our convex training approach outputs an attention map that is close to the ground truth map.

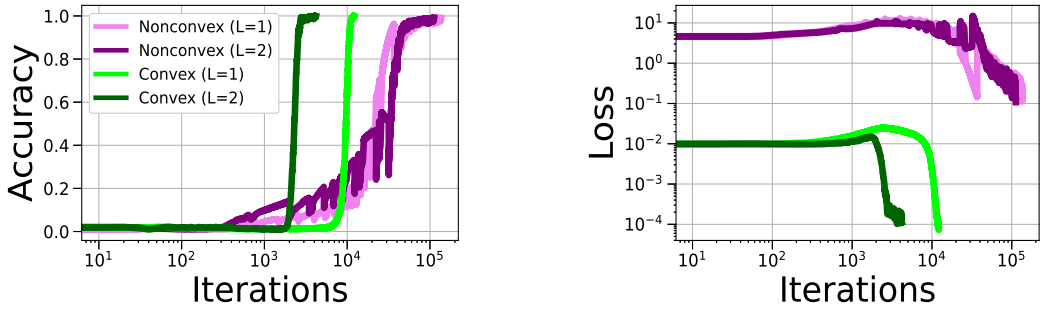
in all the experiments. In Figure 2, we plot the objective values (i.e. training loss + regularization term) and test losses with respect to time in seconds using the data extracted from the sixth layer of pretrained BERT model. We observe that our convex training approach achieves almost an order of magnitude smaller objective value than the standard nonconvex training, which is possibly stuck at a local minimum. This effectiveness in training also translates into better generalization, i.e., our convex training approach obtains a lower test loss than the standard nonconvex training. In order to understand the functions learned by each models, we also analyzed the attention maps in Figure 3. Here, we show that standard nonconvex training fails to learn the underlying model and outputs a uniform attention map across token. However, our convex training outputs an attention map that is quite similar to the ground truth attention map, and therefore we successfully learn the structure in the training data. Hence, these experiments clearly illustrate the effectiveness of our convex training approach in both training and testing.

Algorithmic datasets and Grokking: Inspired by the grokking phenomenon observed in Power et al. (2022), we next validate the effectiveness of our convex training approach on algorithmic datasets. Particularly, we use the same setting in Power et al. (2022), and evaluate the performance on modular division operations with $\mod 97$ and $\mod 15$, where we train the architectures till they reach 99% test accuracy whenever possible. In Figure 4, we first replicate the results in Power et al. (2022) and confirm that the grokking phenomenon indeed emerges here, i.e., the nonconvex curve (purple) reaches 100% training accuracy at around 10^3 iterations in Figure 4a while it requires more than 10^5 iterations to reach perfect generalization in Figure 4b. We also compare the nonconvex and convex training approaches. Here, we show that our convex training approach converges to the perfect generalization accuracy $10\times$ faster than the nonconvex one in Figure 4b. Moreover, the convex model also yield significantly lower test loss in Figure 4c, which implies that it has higher confidence in test predictions and therefore more robust than standard nonconvex training.



(a) Training accuracy (b) Test accuracy (c) Test loss

Figure 4: Comparison of the convex and nonconvex models on the modular division operation $\bmod 97$. Here, we train the networks to reach 99% test accuracy and show that our convex training approach exhibits a significantly faster convergence and lower test loss than the nonconvex training.

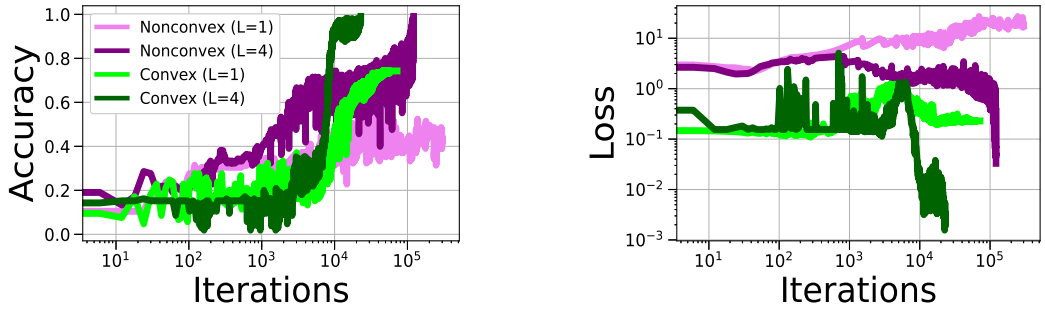


(a) Test accuracy (b) Test loss

Figure 5: Comparison of one- and two-layer transformer networks on the modular division task $\bmod 97$, where L denotes the number of layers in each model. We observe that introducing one more layer substantially improves the convergence speed of our convex formulation while it fails to make a noticeable impact on the nonconvex formulation.

We remark that in the previous section, we theoretically analyze only single attention/transformer blocks. However, since the benign impact of depth or number of layers (denoted as L) has already been empirically proven in the deep learning literature, we also propose an extension of our convex model to deeper settings. We basically stack the convex transformer layers in (11) to obtain an arbitrarily deep network. In Figure 5, we compare the performance of two-layer transformer networks with one-layer networks. Here, we observe that while adding one more layer results in significant improvements for the convex model, especially in terms of optimization speed, it fails to make any discernible difference for the nonconvex model. Moreover, we run the algorithms on the $\bmod 15$ operation which is basically more challenging task due to smaller number of samples. In this case, one-layer models are not able learn the underlying task perfectly as demonstrated in Figure 6 but our convex model is significantly better in terms of both test accuracy and test loss. By increasing the number of layers to four, we enable both models to achieve perfect generalization accuracy. Our deep model reaches this level much faster and also yields lower test loss than the nonconvex model.

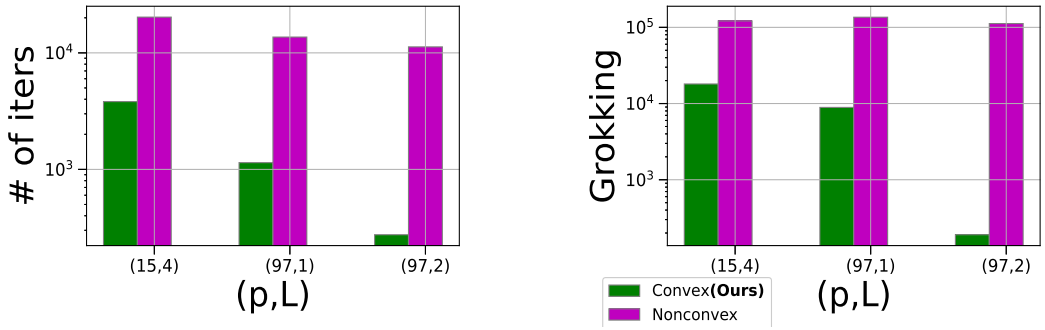
We next empirically analyze the grokking phenomenon on both our convex and standard nonconvex models. For this purpose, we plot the number of iterations to reach 99% test accuracy for each of our experiments in Figure 7a. Notice that here we do not include the one-layer results for the $\bmod 15$ case, since both models fail to achieve perfect generalization in that case. Figure 7a clearly shows that our convex training approach converges to the 99% accuracy level substantially faster than the standard nonconvex training. Therefore, we also mitigate the impact of the grokking phenomenon as demonstrated in Figure 7b, where we quantify the amount of grokking in terms of the number of iterations. Based on this experiment, we also conjecture that the grokking phenomenon can be mostly attributed to the highly nonlinear and nonconvex structure of standard transformer models.



(a) Training accuracy

(b) Test loss

Figure 6: Comparison of one- and four-layer transformer networks on the modular division task $\bmod 15$. Here, one-layer networks fail to achieve 99% test accuracy however our convex training approach (light green) still generalizes better than the nonconvex training (light purple). We also show that perfect generalization in terms of accuracy can be achieved with four-layer networks.



(a) Training iterations

(b) Grokking iterations

Figure 7: Amount of grokking in terms of # of iterations required by our convex and standard nonconvex training approaches, where p and L denote the coefficient for the modular division and the number of layers, respectively. Here, we do not include the one-layer results in Figure 6 since both algorithms fail to achieve 99% test accuracy. We demonstrate that the impact of grokking is substantially mitigated with our convex training approach.

5 Conclusion

In this paper, we studied the regularized training problem of attention/transformer networks and developed a convex analytic framework to train these networks. Particularly, we first proposed a convex alternative to the self-attention mechanism and then reformulated the training problem with this alternative attention mechanism as convex optimization problems. Thanks to our convex reformulation, we globally optimize the network parameters without requiring any kind of nonconvex optimization heuristics. In addition, the functions learned by our reformulation is transparent and interpretable. More importantly, the reformulated problem reveals a sparsity-inducing regularization mechanism across tokens in the data, which also sheds more light on the structure of the resulting function and its generalization properties. We then empirically verified effectiveness of our convex training approach over standard nonconvex training via several numerical experiments.

We also note that analyzing transformer networks through the lens of convex optimization theory is extremely crucial since it may result in substantial improvements in the understanding and optimization of these networks. However, it is also quite challenging due to the inherent nonconvex structure of the network model. To the best of our knowledge, this paper is the first step in this direction and therefore has some limitations which can hopefully be eliminated by future work. Specifically, in this paper, we mainly focused on the theory side of convex analysis and empirically validated the theory on a few small-scale problem instances. We hope that a comprehensive and large-scale empirical verification of our theory will be conducted by the follow-up papers.

REFERENCES

Sergey Bakin et al. Adaptive regression and model selection in data mining problems. 1999.

Boaz Barak, Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *arXiv preprint arXiv:2207.08799*, 2022.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth, 2021. URL https://arxiv.org/abs/2103.03404.

Tolga Ergen and Mert Pilanci. Convex geometry and duality of over-parameterized neural networks. *Journal of machine learning research*, 2021.

Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? *arXiv preprint arXiv:2109.04553*, 2021.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. Lm-debugger: An interactive tool for inspection and intervention in transformer-based language models. *arXiv preprint arXiv: Arxiv-2204.12130*, 2022a.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv: Arxiv-2203.14680*, 2022b.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Bingyuan Liu, Yogesh Balaji, Lingzhou Xue, and Martin Renqiang Min. Analyzing attention mechanisms through lens of sample complexity and loss landscape, 2021a. URL https://openreview.net/forum?id=8KhxoxKP3iL.

Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps, 2021b.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv: Arxiv-2202.05262*, 2022.

Aaron Mishkin, Arda Sahiner, and Mert Pilanci. Fast convex optimization for two-layer relu networks: Equivalent model classes and cone decompositions. *arXiv preprint arXiv:2202.01331*, 2022.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7695–7705. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/pilanci20a.html.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34, 2021.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pp. 2667–2690. PMLR, 2019.

Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. On layer normalizations and residual connections in transformers. *arXiv preprint arXiv:2206.00330*, 2022.

Yuki Tatsunami and Masato Taki. Raftmlp: Do mlp-based models dream of winning over computer vision? *arXiv preprint arXiv:2108.04384*, 2021.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. Attention interpretability across nlp tasks, 2019. URL https://arxiv.org/abs/1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *arXiv preprint arXiv:2111.11418*, 2021.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Appendix

Table of Contents

A Proofs of the results in the main paper **12**

A.1 Proof of Lemma 1 12

A.2 Proof of Proposition 1 13

A.3 Proof of Theorem 1 14

A.4 Proof of Theorem 2 15

A.5 Proof of Theorem 3 17

A Proofs of the results in the main paper

A.1 Proof of Lemma 1

We first note that similar scaling techniques were previously studied in several papers, e.g., Lemma 1 of Ergen & Pilanci (2021), Theorem 1 of Neyshabur et al. (2014), Section 2 of Savarese et al. (2019), equation (2-3) of Pilanci & Ergen (2020).

We start with restating the optimization problem as follows

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}_{2j}\in\mathbb{R}^d, w_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, y_i\right) + \frac{\beta}{2}\sum_{j=1}^{h}\left(\|\mathbf{w}_{2j}\|_2^2 + (w_{3j})^2\right), \tag{12}$$

We first apply the following scaling for $\{\mathbf{w}_{2j}, w_{3j}\}_{j=1}^m$

$$\bar{\mathbf{w}}_{2j} := \alpha_j \mathbf{w}_{2j},\ \bar{w}_{3j} := \frac{w_{3j}}{\alpha_j}. \tag{13}$$

where $\alpha_j > 0$. Since this scaling doesn't change the output of the network, i.e.,

$$\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \bar{\mathbf{w}}_{2j} \bar{w}_{3j} = \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j},$$

the training loss part of the objective function stays the same. Thus, we can directly search for the optimal scaling parameter $\alpha_j > 0$ by minimizing the regularization term via the following AM-GM inequality

$$\sum_{j=1}^{h}\left(\|\bar{\mathbf{w}}_{2j}\|_2^2 + (\bar{w}_{3j})^2\right) = \sum_{j=1}^{h}\left(\alpha_j^2 \|\mathbf{w}_{2j}\|_2^2 + \frac{(w_{3j})^2}{\alpha_j^2}\right) \geq 2\sum_{j=1}^{h}\left(\|\mathbf{w}_{2j}\|_2 |w_{3j}|\right) = 2\sum_{j=1}^{h}\left(\|\bar{\mathbf{w}}_{2j}\|_2 |\bar{w}_{3j}|\right)$$

where the equality is achieved when $\alpha_j = \sqrt{\frac{|w_{3j}|}{\|\mathbf{w}_{2j}\|_2}}$. Thus, we obtain a reformulation of (12) where the regularization term is in a multiplicative form as follows

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}_{2j}\in\mathbb{R}^d, w_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, y_i\right) + \beta\sum_{j=1}^{h} \|\mathbf{w}_{2j}\|_2 |w_{3j}|. \tag{14}$$

Next, we apply a variable change to the reformulation in (14) as follows

$$\mathbf{w}_{2j}' := \frac{\mathbf{w}_{2j}}{\|\mathbf{w}_{2j}\|_2}, \quad w_{3j}' := w_{3j}\|\mathbf{w}_{2j}\|_2.$$

With this variable change, we rewrite (14) as

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}'_{2j}:\left\|\mathbf{w}'_{2j}\right\|_2=1 \\ w'_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^{\top}\mathbf{X}_i\mathbf{w}'_{2j}w'_{3j}, y_i\right) + \beta\sum_{j=1}^{h}|w'_{3j}|. \tag{15}$$

This concludes the proof and yields the following equivalent formulation of (15)

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}'_{2j}:\left\|\mathbf{w}'_{2j}\right\|_2=1 \\ w'_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^{\top}\mathbf{X}_i\mathbf{w}'_{2j}w'_{3j}, y_i\right) + \beta\left\|\mathbf{w}'_3\right\|_1.$$

We also note that the equality constraint $\left\|\mathbf{w}'_{2j}\right\|_2 = 1$ can be relaxed as $\left\|\mathbf{w}'_{2j}\right\|_2 \leq 1$ due to the optimality conditions arising from the regularization term $\left\|\mathbf{w}'_3\right\|_1$. □

A.2 Proof of Proposition 1

Given an optimal solution to (7), denoted as $\mathbf{Z}^* \in \mathbb{R}^{n\times d}$, we first rewrite this solution as a summation of rank-1 matrices as follows

$$\mathbf{Z}^* = \sum_{j=1}^{h}\mathbf{e}_j\mathbf{z}_j^{\top} = \sum_{j=1}^{h}\mathbf{e}_j\frac{\mathbf{z}_j^{\top}}{\sqrt{\left\|\mathbf{z}_j\right\|_2}}\sqrt{\left\|\mathbf{z}_j\right\|_2}$$

where $\mathbf{e}_j \in \mathbb{R}^n$ is the jth ordinary basis vector and we assume that there are h nonzero rows out of n rows of $\mathbf{Z}$ due to the sparsity-inducing regularization in (7). Then, this implies that the output of the optimal can be equivalently formulated as follows

$$\begin{aligned}\operatorname{trace}\left(\mathbf{Z}^{*\top}\mathbf{X}_i\right) &= \sum_{j=1}^{h}\mathbf{e}_j^{\top}\mathbf{X}_i\frac{\mathbf{z}_j}{\sqrt{\left\|\mathbf{z}_j\right\|_2}}\sqrt{\left\|\mathbf{z}_j\right\|_2} \\ &= \sum_{j=1}^{h}\mathbf{w}_{1j}^{*\top}\mathbf{X}_i\mathbf{w}_{2j}^{*}w_{3j}^{*}\end{aligned} \implies \mathbf{w}_{1j}^{*} = \mathbf{e}_j,\ \mathbf{w}_{2j}^{*} = \frac{\mathbf{z}_j}{\sqrt{\left\|\mathbf{z}_j\right\|_2}},\ w_{3j}^{*} = \sqrt{\left\|\mathbf{z}_j\right\|_2},$$

where $\{\mathbf{w}_{1j}^{*}, \mathbf{w}_{2j}^{*}, w_{3j}^{*}\}_{j=1}^{h}$ denotes an optimal solution to (5).

Next, we show that both of these solution sets achieve the same objective value

$$\begin{aligned}f\left(\{\mathbf{X}_i, y_i\}_{i=1}^{N}\right) &:= \sum_{i=1}^{N}\mathcal{L}\left(\sum_{j=1}^{h}\mathbf{w}_{1j}^{*\top}\mathbf{X}_i\mathbf{w}_{2j}^{*}w_{3j}^{*}, y_i\right) + \frac{\beta}{2}\sum_{j=1}^{h}\left(\left\|\mathbf{w}_{2j}^{*}\right\|_2^2 + \left(w_{3j}^{*}\right)^2\right) \\ &= \sum_{i=1}^{N}\mathcal{L}\left(\sum_{j=1}^{h}\mathbf{e}_j^{\top}\mathbf{X}_i\frac{\mathbf{z}_j}{\sqrt{\left\|\mathbf{z}_j\right\|_2}}\sqrt{\left\|\mathbf{z}_j\right\|_2}, y_i\right) + \frac{\beta}{2}\sum_{j=1}^{h}\left(\left\|\frac{\mathbf{z}_j}{\sqrt{\left\|\mathbf{z}_j\right\|_2}}\right\|_2^2 + \left(\sqrt{\left\|\mathbf{z}_j\right\|_2}\right)^2\right) \\ &= \sum_{i=1}^{N}\mathcal{L}\left(\sum_{j=1}^{h}\mathbf{e}_j^{\top}\mathbf{X}_i\mathbf{z}_j, y_i\right) + \frac{\beta}{2}\sum_{j=1}^{h}\left(\left\|\mathbf{z}_j\right\|_2 + \left\|\mathbf{z}_j\right\|_2\right) \\ &= \sum_{i=1}^{N}\mathcal{L}\left(\sum_{j=1}^{h}\operatorname{trace}\left(\mathbf{z}_j\mathbf{e}_j^{\top}\mathbf{X}_i\right), y_i\right) + \beta\sum_{j=1}^{h}\left\|\mathbf{z}_j\right\|_2 \\ &= \sum_{i=1}^{N}\mathcal{L}\left(\operatorname{trace}\left(\mathbf{Z}^{*\top}\mathbf{X}_i\right), y_i\right) + \beta\sum_{k=1}^{n}\left\|\mathbf{z}_k\right\|_2,\end{aligned} \tag{16}$$

where the last inequality follows from the fact that there are h nonzero rows out of n rows of $\mathbf{Z}$ due to the sparsity-inducing regularization in (7). Note that (16) and (7) are the same objectives evaluated at $\mathbf{Z}^*$, which concludes the proof. □

A.3 PROOF OF THEOREM 1

We first provide a summary of our proof strategy. For the derivations of the convex formulation, we basically need to find the bidual form of (5), i.e., the dual of the dual problem. Thus, we start with taking the dual of (5). To avoid nonconvexity in the dual problem, we reformulate the dual constraint, which makes the problem nonconvex, as a convex constraint. Therefore, we obtain a convex dual problem. Then, we take the dual of the dual problem to get the bidual formulation of (5). Since we convexify the dual problem, the bidual formulation is also a convex problem. Therefore, we achieve an equivalent convex formulation of the original nonconvex training problem (5). We also note that a similar proof strategy was also used in Pilanci & Ergen (2020).

In order to take the dual of (6) (i.e. restated below for the convenience of the reader) we need to form the Lagrangian function for the following optimization problem

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \|\mathbf{w}_{2j}\|_2\leq 1, w_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, y_i\right) + \beta \left\|\mathbf{w}_3\right\|_1 .$$

To construct the Lagrangian function, we first introduce an additional variable $\hat{\mathbf{y}} \in \mathbb{R}^N$ as follows

$$\min_{\substack{\hat{\mathbf{y}}\in\mathbb{R}^N, \mathbf{w}_{1j}\in\Delta \\ \|\mathbf{w}_{2j}\|_2\leq 1, w_{3j}\in\mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\hat{y}_i, y_i\right) + \beta \left\|\mathbf{w}_3\right\|_1 \quad \text{s.t.} \quad \hat{y}_i = \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}, \ \forall i \in [n]. \tag{17}$$

Now we can form the Lagrangian for (17) as

$$\begin{aligned}
L(\mathbf{v}, \mathbf{y}, \mathbf{w}_3) &:= \sum_{i=1}^{N} \mathcal{L}\left(\hat{y}_i, y_i\right) + \beta \left\|\mathbf{w}_3\right\|_1 + \sum_{i=1}^{N} v_i \left(\hat{y}_i - \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}\right) \\
&= \sum_{i=1}^{N} \mathcal{L}\left(\hat{y}_i, y_i\right) + \sum_{i=1}^{N} v_i \hat{y}_i + \beta \left\|\mathbf{w}_3\right\|_1 - \sum_{i=1}^{N} v_i \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j} \\
&= \sum_{i=1}^{N} \mathcal{L}\left(\hat{y}_i, y_i\right) + \sum_{i=1}^{N} v_i \hat{y}_i + \beta \left\|\mathbf{w}_3\right\|_1 - \sum_{j=1}^{h} \sum_{i=1}^{N} v_i \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} w_{3j}
\end{aligned}$$

Minimizing the Lagrangian $L(\cdot)$ yields the following dual problem of (5)

$$\max_{\mathbf{v}\in\mathbb{R}^N} -\mathcal{L}^*\left(\mathbf{v}, \mathbf{y}\right) \quad \text{s.t.} \max_{\mathbf{w}_1\in\Delta, \|\mathbf{w}_2\|_2\leq 1} \left|\sum_{i=1}^{N} v_i \mathbf{w}_1^\top \mathbf{X}_i \mathbf{w}_2\right| \leq \beta, \tag{18}$$

where $\mathcal{L}^*\left(\cdot\right)$ denotes the Fenchel congregate function of the original loss function $\mathcal{L}\left(\cdot\right)$ (Boyd & Vandenberghe, 2004), which is defined as follows

$$\mathcal{L}^*\left(\mathbf{v}, \mathbf{y}\right) := \max_{\mathbf{z}\in\mathbb{R}^N} \mathbf{z}^\top \mathbf{v} - \mathcal{L}\left(\mathbf{z}, \mathbf{y}\right).$$

In order to convexify the dual constraint, we next find the maximizers of the dual constraint as follows

$$\begin{aligned}
\max_{\mathbf{w}_1\in\Delta, \|\mathbf{w}_2\|_2\leq 1} \left|\sum_{i=1}^{N} v_i \mathbf{w}_1^\top \mathbf{X}_i \mathbf{w}_2\right| &= \max_{\mathbf{w}_1\in\Delta} \left\|\sum_{i=1}^{N} v_i \mathbf{w}_1^\top \mathbf{X}_i\right\|_2 \\
&= \max_{\mathbf{w}_1\in\Delta} \left\|\sum_{i=1}^{N} \sum_{k=1}^{n} v_i w_{1k} \mathbf{x}_{ik}\right\|_2 \\
&\leq \max_{\mathbf{w}_1\in\Delta} \sum_{k=1}^{n} w_{1k} \left\|\sum_{i=1}^{N} v_i \mathbf{x}_{ik}\right\|_2 \\
&= \max_{k\in[n]} \left\|\sum_{i=1}^{N} v_i \mathbf{x}_{ik}\right\|_2 ,
\end{aligned} \tag{19}$$

where the upperbound is achieved when each $\mathbf{w}_1$ has is a vector of zeros except a single one located at the index of maximum norm of weighted tokens.

Based on the observation in (19), we can equivalently write the dual problem in (18) as follows

$$\begin{aligned} &\max_{\mathbf{v}\in\mathbb{R}^N} -\mathcal{L}^*\left(\mathbf{v},\mathbf{y}\right) \quad \text{s.t.} \max_{k\in[h]} \left\| \sum_{i=1}^N v_i \mathbf{x}_{ik} \right\|_2 \leq \beta \\ &= \max_{\mathbf{v}\in\mathbb{R}^N} -\mathcal{L}^*\left(\mathbf{v},\mathbf{y}\right) \quad \text{s.t.} \left\| \sum_{i=1}^N v_i \mathbf{x}_{ik} \right\|_2 \leq \beta, \forall k \in [n]. \end{aligned} \tag{20}$$

Next, we form the Lagrangian for the dual problem (20)

$$L(\mathbf{v},\mathbf{y},\boldsymbol{\lambda}) := -\mathcal{L}^*\left(\mathbf{v},\mathbf{y}\right) + \sum_{k=1}^n \lambda_k \left(\beta - \left\| \sum_{i=1}^N v_i \mathbf{x}_{ik} \right\|_2 \right)$$

and the corresponding optimization problem can be written in terms of the Lagrangian as

$$\max_{\mathbf{v}\in\mathbb{R}^N} \min_{\boldsymbol{\lambda}\geq\mathbf{0}} L(\mathbf{v},\mathbf{y},\boldsymbol{\lambda}) = -\frac{1}{2}\left\|\mathbf{v}-\mathbf{y}\right\|_2^2 + \frac{1}{2}\left\|\mathbf{y}\right\|_2^2 + \sum_{k=1}^n \lambda_k \left(\beta - \left\| \sum_{i=1}^N v_i \mathbf{x}_{ik} \right\|_2 \right).$$

Then, we introduce additional variables $\mathbf{r}_k \in \mathbb{R}^d$ to equivalently formulate the optimization problem above as

$$\max_{\mathbf{v}\in\mathbb{R}^n} \min_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\mathbf{r}_k:\|\mathbf{r}_k\|_2\leq 1} -\mathcal{L}^*\left(\mathbf{v},\mathbf{y}\right) + \sum_{k=1}^n \lambda_k \left(\beta - \mathbf{r}_k^\top \sum_{i=1}^N v_i \mathbf{x}_{ik} \right).$$

Due to Sion's minimax theorem (Sion, 1958), we can change the order the minimization and maximization to obtain closed-form solutions for the maximization over the dual variable $\mathbf{v}$. This yields the following problem

$$\min_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\mathbf{r}_k:\|\mathbf{r}_k\|_2\leq 1} \sum_{i=1}^N \mathcal{L}\left(\sum_{k=1}^n \lambda_k \mathbf{r}_k^\top \mathbf{x}_{ik}, y_i \right) + \beta \sum_{k=1}^n \lambda_k.$$

Next, we apply a variable change as $\mathbf{z}_k := \lambda_k \mathbf{r}_k$, then the problem above reduces to

$$\min_{\mathbf{z}_k:\|\mathbf{z}_k\|_2\leq\lambda_k} \sum_{i=1}^N \mathcal{L}\left(\sum_{k=1}^n \mathbf{z}_k^\top \mathbf{x}_{ik}, y_i \right) 2 + \beta \sum_{k=1}^n \lambda_k.$$

From the KKT conditions, we now that $\lambda_k = \left\|\mathbf{z}_k\right\|_2$ at a global optimum, therefore, the problem can be reformulated as

$$\min_{\mathbf{z}_k} \sum_{i=1}^N \mathcal{L}\left(\sum_{k=1}^n \mathbf{z}_k^\top \mathbf{x}_{ik}, y_i \right) + \beta \sum_{k=1}^n \left\|\mathbf{z}_k\right\|_2,$$

which is the same formulation with (7) and therefore concludes the proof. □

A.4 Proof of Theorem 2

We first apply the scaling technique in Lemma 1 for $\{\mathbf{w}_{2j}, \mathbf{w}_{3j}\}_{j=1}^m$

$$\bar{\mathbf{w}}_{2j} := \alpha_j \mathbf{w}_{2j}, \;\; \bar{\mathbf{w}}_{3j} := \frac{\mathbf{w}_{3j}}{\alpha_j}.$$

Then, following the same steps in Lemma 1, (8) can be equivalently formulated as

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}_{2j}:\|\mathbf{w}_{2j}\|_2\leq 1 \\ \mathbf{w}_{3j}\in\mathbb{R}^e}} \sum_{i=1}^N \mathcal{L}\left(\sum_{j=1}^h \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} \mathbf{w}_{3j}, \mathbf{y}_i \right) + \frac{\beta}{2} \sum_{j=1}^h \left\|\mathbf{w}_{3j}\right\|_1. \tag{21}$$

Next, we again construct the Lagrangian function by introducing an additional variable $\hat{\mathbf{y}}_i \in \mathbb{R}^c, \forall i \in [N]$ as follows

$$\min_{\substack{\hat{\mathbf{y}}_i \in \mathbb{R}^c, \mathbf{w}_{1j} \in \Delta \\ \|\mathbf{w}_{2j}\|_2 \leq 1, w_{3j} \in \mathbb{R}}} \sum_{i=1}^{N} \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \beta \sum_{j=1}^{h} \|\mathbf{w}_{3j}\|_1 \quad \text{s.t.} \quad \hat{\mathbf{y}}_i = \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} \mathbf{w}_{3j}, \ \forall i \in [N]. \tag{22}$$

Now we can form the Lagrangian for (22) as

$$\begin{aligned} L\left(\{\mathbf{v}_i\}_{i=1}^N, \mathbf{y}, \{\mathbf{w}_{3j}\}_{j=1}^h\right) &:= \sum_{i=1}^{N} \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \beta \sum_{j=1}^{h} \|\mathbf{w}_{3j}\|_1 + \sum_{i=1}^{N} \mathbf{v}_i^\top \left(\hat{\mathbf{y}}_i - \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} \mathbf{w}_{3j}\right) \\ &= \sum_{i=1}^{N} \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \sum_{i=1}^{N} \mathbf{v}_i^\top \hat{\mathbf{y}}_i + \beta \sum_{j=1}^{h} \|\mathbf{w}_{3j}\|_1 - \sum_{i=1}^{N} \mathbf{v}_i^\top \sum_{j=1}^{h} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} \mathbf{w}_{3j} \\ &= \sum_{i=1}^{N} \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \sum_{i=1}^{N} \mathbf{v}_i^\top \hat{\mathbf{y}}_i + \beta \sum_{j=1}^{h} \|\mathbf{w}_{3j}\|_1 - \sum_{j=1}^{h} \sum_{i=1}^{N} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j} \mathbf{v}_i^\top \mathbf{w}_{3j} \end{aligned}$$

Minimizing the Lagrangian $L(\cdot)$ yields the following dual problem of (8)

$$\max_{\mathbf{v}_i \in \mathbb{R}^c} -\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) \quad \text{s.t.} \max_{\mathbf{w}_1 \in \Delta, \|\mathbf{w}_2\|_2 \leq 1} \left\| \sum_{i=1}^{N} \mathbf{v}_i \mathbf{w}_1^\top \mathbf{X}_i \mathbf{w}_2 \right\|_\infty \leq \beta, \tag{23}$$

where $\mathcal{L}^*(\cdot)$ denotes the Fenchel congregate function of the original loss function $\mathcal{L}(\cdot)$ (Boyd & Vandenberghe, 2004), which is defined as follows

$$\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) := \max_{\mathbf{Z} \in \mathbb{R}^{N \times c}} \operatorname{trace}\left(\mathbf{Z}^\top \mathbf{V}\right) - \mathcal{L}\left(\mathbf{Z}, \mathbf{Y}\right),$$

where $\mathbf{V}, \mathbf{Y} \in \mathbb{R}^{N \times c}$ are the matrix representations for the set of variables $\{\mathbf{v}_i, \mathbf{y}_i\}_{i=1}^N$. In order to characterize the optimal layer weight explicitly, we next find the maximizers of the dual constraint as follows

$$\begin{aligned} \max_{\mathbf{w}_1 \in \Delta, \|\mathbf{w}_2\|_2 \leq 1} \left\| \sum_{i=1}^{N} \mathbf{v}_i \mathbf{w}_1^\top \mathbf{X}_i \mathbf{w}_2 \right\|_\infty &= \max_{\mathbf{w}_1 \in \Delta} \max_{l \in [c]} \left\| \sum_{i=1}^{N} v_{il} \mathbf{w}_1^\top \mathbf{X}_i \right\|_2 \\ &= \max_{\mathbf{w}_1 \in \Delta} \max_{l \in [c]} \left\| \sum_{i=1}^{N} \sum_{k=1}^{n} v_{il} w_{1k} \mathbf{x}_{ik} \right\|_2 \\ &\leq \max_{\mathbf{w}_1 \in \Delta} \max_{l \in [c]} \sum_{k=1}^{n} w_{1k} \left\| \sum_{i=1}^{N} v_i \mathbf{x}_{ik} \right\|_2 \\ &= \max_{l \in [c]} \max_{k \in [n]} \left\| \sum_{i=1}^{N} v_{il} \mathbf{x}_{ik} \right\|_2, \end{aligned} \tag{24}$$

where the upperbound is achieved when each $\mathbf{w}_1$ has is a vector of zeros except a single one located at the index of maximum norm of weighted tokens.

Based on the observation in (24), we can equivalently write the dual problem in (23) as follows

$$\max_{\mathbf{v} \in \mathbb{R}^N} -\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) \quad \text{s.t.} \left\| \sum_{i=1}^{N} v_{il} \mathbf{x}_{ik} \right\|_2 \leq \beta, \forall k \in [n], \forall l \in [c]. \tag{25}$$

Then directly following the steps in the proof of Theorem 1 yields the following convex optimization problem

$$\min_{\mathbf{Z}_l \in \mathbb{R}^{n \times d}} \sum_{i=1}^{N} \sum_{l=1}^{c} \mathcal{L}\left(\operatorname{trace}\left(\mathbf{Z}_l^\top \mathbf{X}_i\right), y_{il}\right) + \beta \sum_{l=1}^{c} \sum_{k=1}^{n} \|\mathbf{z}_{lk}\|_2.$$

□

A.5 PROOF OF THEOREM 3

We first apply the scaling technique in Lemma 1 for $\{\mathbf{w}_{2j}, \mathbf{w}_{3j}\}_{j=1}^m$

$$\bar{\mathbf{w}}_{2j} := \alpha_j \mathbf{w}_{2j}, \ \bar{\mathbf{w}}_{3j} := \frac{\mathbf{w}_{3j}}{\alpha_j}.$$

Then, following the same steps in Lemma 1, (10) can be equivalently formulated as

$$\min_{\substack{\mathbf{w}_{1j}\in\Delta \\ \mathbf{w}_{2j}:\|\mathbf{w}_{2j}\|_2\leq 1 \\ \mathbf{w}_{3j}\in\mathbb{R}^c}} \sum_{i=1}^N \mathcal{L}\left(\sum_{j=1}^h \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\mathbf{w}_{3j}, \mathbf{y}_i\right) + \frac{\beta}{2}\sum_{j=1}^h \|\mathbf{w}_{3j}\|_1 . \tag{26}$$

Next, we again construct the Lagrangian function by introducing an additional variable $\hat{\mathbf{y}}_i \in \mathbb{R}^c, \forall i \in [N]$ as follows

$$\min_{\substack{\hat{\mathbf{y}}_i\in\mathbb{R}^c, \mathbf{w}_{1j}\in\Delta \\ \|\mathbf{w}_{2j}\|_2\leq 1, w_{3j}\in\mathbb{R}}} \sum_{i=1}^N \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \beta\sum_{j=1}^h \|\mathbf{w}_{3j}\|_1 \quad \text{s.t.} \quad \hat{\mathbf{y}}_i = \sum_{j=1}^h \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\mathbf{w}_{3j}, \ \forall i \in [N]. \tag{27}$$

Now we can form the Lagrangian for (27) as

$$\begin{aligned}
L\left(\{\mathbf{v}_i\}_{i=1}^N, \mathbf{y}, \{\mathbf{w}_{3j}\}_{j=1}^h\right) &:= \sum_{i=1}^N \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \beta\sum_{j=1}^h \|\mathbf{w}_{3j}\|_1 + \sum_{i=1}^N \mathbf{v}_i^\top \left(\hat{\mathbf{y}}_i - \sum_{j=1}^h \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\mathbf{w}_{3j}\right) \\
&= \sum_{i=1}^N \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \sum_{i=1}^N \mathbf{v}_i^\top \hat{\mathbf{y}}_i + \beta\sum_{j=1}^h \|\mathbf{w}_{3j}\|_1 - \sum_{i=1}^N \mathbf{v}_i^\top \sum_{j=1}^h \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\mathbf{w}_{3j} \\
&= \sum_{i=1}^N \mathcal{L}\left(\hat{\mathbf{y}}_i, \mathbf{y}_i\right) + \sum_{i=1}^N \mathbf{v}_i^\top \hat{\mathbf{y}}_i + \beta\sum_{j=1}^h \|\mathbf{w}_{3j}\|_1 - \sum_{j=1}^h \sum_{i=1}^N \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\mathbf{v}_i^\top \mathbf{w}_{3j}
\end{aligned}$$

Minimizing the Lagrangian $L(\cdot)$ yields the following dual problem of (10)

$$\max_{\mathbf{v}_i\in\mathbb{R}^c} -\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) \quad \text{s.t.} \max_{\mathbf{w}_{1j}\in\Delta, \|\mathbf{w}_{2j}\|_2\leq 1} \left\|\sum_{i=1}^N \mathbf{v}_i \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)\right\|_\infty \leq \beta, \forall j \in [h], \tag{28}$$

where $\mathcal{L}^*(\cdot)$ denotes the Fenchel congregate function of the original loss function $\mathcal{L}(\cdot)$ (Boyd & Vandenberghe, 2004), which is defined as follows

$$\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) := \max_{\mathbf{Z}\in\mathbb{R}^{N\times c}} \text{trace}\left(\mathbf{Z}^\top \mathbf{V}\right) - \mathcal{L}\left(\mathbf{Z}, \mathbf{Y}\right),$$

where $\mathbf{V}, \mathbf{Y} \in \mathbb{R}^{N\times c}$ are the matrix representations for the set of variables $\{\mathbf{v}_i, \mathbf{y}_i\}_{i=1}^N$.

We next note that we utilize the gated ReLU nonlinearity introduced in Mishkin et al. (2022). Thus, the activations $\sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right)$ can be expressed as

$$\sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right) := \mathbb{1}_{ij} \mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j},$$

where $\mathbb{1}_{ij} := \mathbb{1}\left\{\mathbf{u}_{1j}^\top \mathbf{X}_i \mathbf{u}_{2j} \geq 0\right\}$ and here $\{\mathbf{u}_{1j}, \mathbf{u}_{2j}\}_{j=1}^h$ are fixed vectors that can be randomly selected. For instance, a common choice is $\mathbf{u}_{1j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and $\mathbf{u}_{2j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. For the rest of the derivations, we use this equivalent formulation of the activation function.

In order to characterize the optimal layer weight explicitly, we next find the maximizers of the dual constraint as follows

$$\begin{aligned}
\max_{j\in[h]} \max_{\mathbf{w}_{1j}\in\Delta, \|\mathbf{w}_{2j}\|_2\leq 1} \left\| \sum_{i=1}^{N} \mathbf{v}_i \sigma\left(\mathbf{w}_{1j}^\top \mathbf{X}_i \mathbf{w}_{2j}\right) \right\|_\infty &= \max_{j\in[h]} \max_{l\in[c]} \max_{\mathbf{w}_{1j}\in\Delta} \left\| \sum_{i=1}^{N} v_{il} \mathbb{1}_{ij} \mathbf{w}_{1j}^\top \mathbf{X}_i \right\|_2 \\
&= \max_{j\in[h]} \max_{\mathbf{w}_{1j}\in\Delta} \max_{l\in[c]} \left\| \sum_{i=1}^{N} \sum_{k=1}^{n} v_{il} \mathbb{1}_{ij} w_{1jk} \mathbf{x}_{ik} \right\|_2 \\
&\leq \max_{j\in[h]} \max_{\mathbf{w}_{1j}\in\Delta} \max_{l\in[c]} \sum_{k=1}^{n} w_{1jk} \left\| \sum_{i=1}^{N} v_i \mathbb{1}_{ij} \mathbf{x}_{ik} \right\|_2 \\
&= \max_{j\in[h]} \max_{l\in[c]} \max_{k\in[n]} \left\| \sum_{i=1}^{N} v_{il} \mathbb{1}_{ij} \mathbf{x}_{ik} \right\|_2 , \qquad (29)
\end{aligned}$$

where the upperbound is achieved when each $\mathbf{w}_{1j}$ has is a vector of zeros except a single one located at the index of maximum norm of weighted tokens.

Based on the observation in (29), we can equivalently write the dual problem in (28) as follows

$$\max_{\mathbf{v}\in\mathbb{R}^N} -\mathcal{L}^*\left(\{\mathbf{v}_i\}_{i=1}^N, \{\mathbf{y}_i\}_{i=1}^N\right) \quad \text{s.t.} \left\| \sum_{i=1}^{N} v_{il} \mathbb{1}_{ij} \mathbf{x}_{ik} \right\|_2 \leq \beta, \forall k\in[n], \forall l\in[c], \forall j\in[h]. \qquad (30)$$

Then directly following the steps in the proof of Theorem 2 yields the following convex optimization problem

$$\min_{\mathbf{Z}_{jl}\in\mathbb{R}^{n\times d}} \sum_{i=1}^{N} \sum_{l=1}^{c} \mathcal{L}\left(\sum_{j=1}^{h} \mathbb{1}_{ij} \text{trace}\left(\mathbf{Z}_{jl}^\top \mathbf{X}_i\right), y_{il} \right) + \beta \sum_{l=1}^{c} \sum_{j=1}^{h} \sum_{k=1}^{n} \|\mathbf{z}_{jlk}\|_2 .$$

□