Abstract

Vision transformers using self-attention or its proposed alternatives have demonstrated promising results in many image related tasks. However, the underpinning inductive bias of attention is not well understood. To address this issue, this paper analyzes attention through the lens of convex duality. For the non-linear dot-product self-attention, and alternative mechanisms such as MLP-mixer and Fourier Neural Operator (FNO), we derive equivalent finite-dimensional convex problems that are interpretable and solvable to global optimality. The convex programs lead to block nuclear-norm regularization that promotes low rank in the latent feature and token dimensions. In particular, we show how self-attention networks implicitly clusters the tokens, based on their latent similarity. We conduct experiments for transferring a pre-trained transformer backbone for CIFAR-100 classification by fine-tuning a variety of convex attention heads. The results indicate the merits of the bias induced by attention compared with the existing MLP or linear heads.

1. Introduction

Transformers have recently delivered tremendous success for representation learning in language and vision. This is primarily due to the attention mechanism that effectively mixes the tokens’ representation over the layers to learn the semantics present in the input\(^1\). After the inception of dot-product self-attention (Vaswani et al., 2017), there have been several efficient alternatives that scale nicely with the sequence size for large pretraining tasks; see e.g., (Wang et al., 2020; Shen et al., 2021; Kitaev et al., 2020; Panahi et al., 2021; Xiong et al., 2021). However, the learnable inductive bias of attention is not explored well. A strong theoretical understanding of attention’s inductive bias can motivate designing more efficient architectures, and can explain the generalization ability of these networks.

Self-attention was the fundamental building block in the first proposed vision transformer (ViT) (Dosovitskiy et al., 2020). It consists of an outer product of two linear functions, followed by a non-linearity and a product with another linear function, which makes it non-convex and non-interpretable. One approach to understand attention has been to design new alternatives to self-attention which perform similarly well, which may help explain its underlying mechanisms. One set of work pertains to Multi-Layer Perceptron (MLP) based architectures, (Tolstikhin et al., 2021; Tatsunami & Taki, 2021; Touvron et al., 2021; Liu et al., 2021; Yu et al., 2021), while another line of work proposes Fourier based models (Lee-Thorp et al., 2021; Rao et al., 2021; Li et al., 2020; Guibas et al., 2021). Others have proposed replacing self-attention with matrix decomposition (Geng et al., 2021). While all of these works have appealing applications that leverage general concepts about the structure of attention, they lack any fine-tuned theoretical analysis on these architectures from an optimization perspective.

To address this shortcoming, we leverage convex duality to analyze a single block of self-attention with ReLU activation. Since self-attention incurs quadratic complexity in the sequence, we alternatively analyze more efficient modules. As representatives for more efficient modules, we focus on MLP Mixers (Tolstikhin et al., 2021) and Fourier Neural Operators (FNO) (Li et al., 2020). MLP-mixer mixes tokens (purely) using MLP projections in both the token and feature dimensions. In contrast, FNO mixes tokens using circular convolution that is efficiently implemented based on 2D Fourier transforms.

We find that all three of these analyzed modules are equivalent to finite-dimensional convex optimization problems, indicating that there are guarantees to provably optimize them to their global optima. Furthermore, we make novel observations about the bias induced by the convex models. In particular, convexified equivalents of both self-attention

---

\(^1\)Department of Electrical Engineering, Stanford University, Stanford, CA, USA \(^2\)NVIDIA Corporation, Santa Clara, CA, USA. Correspondence to: Arda Sahiner <sahiner@stanford.edu>.
Unraveling Attention via Convex Duality

Figure 1. Illustration of the implicit convexity of (linear activation) multi-head self-attention. (a) Input image is first divided into $h w = s$ patches, where each patch is represented by a latent vector of dimension $d$. (b) The (non-convex) scaled dot-product self-attention applies learnable weights $Q_j$, $K_j$, $V_j$ to the patch embeddings $X_i$ as in (10). (c) In the equivalent convex optimization problem for the self-attention training objective (14), the Gram matrix $G_i$ is formed that groups latent features in $B$ different blocks, (d) and accordingly the nuclear norm regularization is imposed on the dual variables $Z^{(b,k,l)} \in \mathbb{R}^{d \times c}$ based on the similarity scores $G^{(b)}_{i} [k, l]$.

and MLP-Mixer modules resemble weighted combinations of MLPs, but with additional degrees of freedom (e.g. higher dimensional induced parameters), and a unique block nuclear norm regularization which ties their individual sub-modules together to encourage global cohesiveness. In contrast, the convexified FNO mixer amounts to circular convolution, while slight modifications to the FNO architecture can induce an equivalent group-wise convolution. We experimentally test and compare these convex attention heads for transfer learning of a pre-trained vision transformer to CIFAR-100 classification upon finetuning a single convex head. We observe that the inductive bias of these attention modules outperforms traditional convex models.

Our main contributions are summarized as follows:

- We provide guarantees that self-attention, MLP-Mixer, and FNO with linear and ReLU activation can be solved to their global optima by demonstrating their equivalence to convex optimization problems.
- By analyzing these equivalent convex programs, we provide interpretability to the optimization objectives of these attention modules.
- Our experiments validate the (convex) vision transformers perform better than baseline convex methods in a transfer learning task.

1.1. Related Work

This work is primarily related to two lines of research.

Interpreting attention. One approach has been to experimentally observe the properties of attention networks to understand them. For example, DINO proposes a contrastive self-supervised learning method for ViTs (Caron et al., 2021). It is observed that learned attention maps preserve semantic regions of the images. Another work compares the alignment across layers of trained ViTs and CNNs, concluding that ViTs have more a uniform representation structure across layers of the network (Raghu et al., 2021). Another work uses a Deep Taylor Decomposition approach to visualize portions of input image leading to a particular ViT prediction (Chefer et al., 2021).

Another approach is to analyze the expressivity of attention networks. One work interprets multi-head self-attention as a Bayesian inference, and provides tools to decide how many heads to use, and how to enforce distinct representations in different heads (An et al., 2020). Other analysis has demonstrated that sparse transformers can universally approximate any function (Yun et al., 2020), that multi-head self-attention is at least as expressive as a convolutional layer (Cordonnier et al., 2019), and that dot-product self-attention is not Lipschitz continuous (Kim et al., 2021).

Convex neural networks. Starting with (Pilanci & Ergen, 2020), there has been a long line of work demonstrating that various ReLU-activation neural network architectures have equivalent convex optimization problems. These include two-layer fully connected, convolutional and vector-output networks (Ergen & Pilanci, 2021a; Sahiner et al., 2020; 2021b; Ergen & Pilanci, 2021b; Wang et al., 2021), as well
as deeper ones (Ergen & Pilanci, 2021c,d), autoencoders (Gupta et al., 2021), networks with Batch Normalization (Ergen et al., 2021), polynomial activations (Bartan & Pilanci, 2021a), quantized networks (Bartan & Pilanci, 2021b), and Wasserstein GANs (Sahiner et al., 2021a). Recent work has also demonstrated how to efficiently optimize the simplest forms of these equivalent convex networks, and incorporate additional constraints for adversarial robustness (Mishkin et al., 2022; Bai et al., 2022). However, none of these works have analyzed the building blocks of transformers, which are a leading method in many state-of-the-art vision and language processing tasks.

2. Preliminaries

In general, we analyze supervised learning problems where input training data \( \{X_i \in \mathbb{R}^{n \times d}\}_{i=1}^n \) are the result of a patch embedding layer, and we have corresponding labels of arbitrary size \( \{Y_i \in \mathbb{R}^{r \times c}\}_{i=1}^n \). For arbitrary convex loss function \( \mathcal{L}(\cdot, \cdot) \), we solve the optimization problem

\[
p^* := \min_\theta \sum_{i=1}^n \mathcal{L}(f_\theta(X_i), Y_i) + \mathcal{R}(\theta) \tag{1}
\]

for some learnable parameters \( \theta \), neural network \( f_\theta(\cdot) \), and regularizer \( \mathcal{R}(\cdot) \). Note that this formulation encapsulates both denoising and classification scenarios: in the classification setting of \( r = 1 \), one can absorb global average pooling into the convex loss \( \mathcal{L} \), whereas if \( r = s \), one can directly use squared loss or other convex loss functions. One may also use this formulation to apply to both supervised and self-supervised learning.

In this paper, we denote \((\cdot)_+ := \max\{0, \cdot\}\) as the ReLU non-linearity. We use superscripts, say \( A^{(i_1,i_2)} \), to denote blocks of matrices, and brackets, say \( A_{[i_1,i_2]} \), to denote elements of matrices, where the arguments refer to row (or block of rows) \( i_1 \) and column (or block of columns) \( i_2 \).

2.1. Implicit Convexity of Linear and ReLU MLPs

Previously, it has been demonstrated that standard two-layer ReLU MLPs are equivalent to convex optimization problems. We briefly describe the relevant background to provide context for much of the analysis in this paper. In particular, we are presented with a network with \( m \) neurons in the hidden layer, weight-decay parameter \( \beta > 0 \), and data \( X \in \mathbb{R}^{n \times d}, Y \in \mathbb{R}^{n \times c} \):

\[
p^*_{RMLP} := \min_{\mathbf{w}_{2j}, \mathbf{w}_{2j}} \mathcal{L} \left( \sum_{j=1}^m (\mathbf{Xw}_{1j})_+ \mathbf{w}^\top_{2j}, Y \right) + \beta \sum_{j=1}^m \left( \|w_{1j}\|_2^2 + \|w_{2j}\|_2^2 \right). \tag{2}
\]

While this problem is non-convex as stated, it has been demonstrated that the objective is equivalent to the solution of an equivalent convex optimization problem, and there is a one-to-one mapping between the two problems’ solutions (Sahiner et al., 2020). In particular, this analysis makes use of hyperplane arrangements, which enumerate all possible activation patterns of the ReLU non-linearity:

\[
D := \{ \text{diag}(1\{Xu \geq 0\}) : u \in \mathbb{R}^d \} \tag{3}
\]

The set \( D \) is clearly finite, and its cardinality is bounded as \( P := |D| \leq 2^r \left( \frac{c(n-1)}{r} \right)^r \), where \( r := \text{rank}(X) \) (Stanley et al., 2004; Pilanci & Ergen, 2020). Through convex duality analysis, we can express an equivalent convex optimization problem by enumerating over the finite set of arrangements \( \{D_j\}_{j=1}^P \). We define the following norm

\[
\|Z\|_{*,K} := \min_{t \geq 0} \text{s.t. } Z \in tC \tag{4}
\]

\[
C := \text{conv}\{Z = uv^\top : Ku \geq 0, \|Z\|_* \leq 1, v \in \mathbb{R}^c\}
\]

This norm is a quasi-nuclear norm, which differs from the standard nuclear norm in that the factorization upon which it relies puts a constraint on its left factors, which in our case will be an affine constraint. In convex ReLU neural networks, \( K \) is chosen to enforce the existence of \( \{u_k, v_k\} \) such that \( Z = \sum_k u_k v_k^\top \) and \( D_j XZ = \sum_k (Xu_k)_+ v_k \), and penalizes \( \sum_k \|u_k\|_2 \|v_k\|_2^2 / 2 \).

With this established, it can be shown that

\[
p^*_{RMLP} = \min_{\{Z_j\}_{j=1}^P} \mathcal{L} (\sum_{j=1}^P D_j XZ_j, Y) + \beta \sum_{j=1}^P \|Z_j\|_{*,K_j} \tag{5}
\]

The two-layer ReLU MLP optimization problem is thus expressed as a piece-wise linear model with a constrained nuclear norm regularization. This contrasts with the two-layer linear activation MLP, whose convex equivalent is

\[
p^*_{LMLP} = \min_{Z} \mathcal{L}(XZ, Y) + \beta \|Z\|_* \tag{6}
\]

This nuclear norm penalty is known to encourage low-rank solutions (Candès & Tao, 2010; Recht et al., 2010) and appears in matrix factorization problems (Gunasekar et al., 2017). One can also define the gated ReLU activation, where the ReLU gates are fixed to some \( \{h_j\}_{j=1}^m \) (Fiat et al., 2019)

\[
g(Xw_{1j}) := \text{diag}(1\{Xh_j \geq 0\}) (Xw_{1j}) \tag{7}
\]

Then, defining \( \{D_j\}_{j=1}^m := \{ \text{diag}(1\{Xh_j \geq 0\}) \}_{j=1}^m \), the corresponding convex gated ReLU activation two-layer network objective follows directly from the ReLU and linear

\[\text{We illustrate an example of } \|Z\|_{*,K} \text{ in the Appendix C.1.}\]
We note that for both linear and gated ReLU formulations, the regularization on the convex weights becomes the standard nuclear norm, since ReLU constraints no longer have to be enforced. It has been demonstrated that there is a small approximation gap between Gated ReLU and ReLU networks, and ReLU networks can be formed from the solutions to Gated ReLU problems (Mishkin et al., 2022).

In terms of efficient algorithms to solve these problems, using an accelerated proximal gradient descent algorithm applied to the convex linear and gated ReLU programs, one can achieve an $\epsilon$-optimal solution in $O(1/\sqrt{\epsilon})$ iterations (Toh & Yun, 2010). For the convex ReLU formulation, (Sahinier et al., 2020) proposes a Frank-Wolfe algorithm for convex MLPs which can be adapted to this case, which in the general case requires $O(1/\epsilon)$ iterations for $\epsilon$-optimality (Jaggi, 2013).

In the subsequent sections, we will demonstrate how common vision transformer blocks with linear and ReLU activations can be related to equivalent convex optimization problems through similar convex duality techniques.\textsuperscript{3}

3. Implicit Convexity of Self-Attention

The canonical Vision Transformer (ViT) uses self-attention and MLPs as its backbone (Dosovitskiy et al., 2020). In particular, a single “head” of a self-attention network is given by the following:

$$f_j(X_i) := \sigma \left( \frac{X_i Q_j K_j^\top X_j}{\sqrt{d}} \right) X_j V_j, \quad (9)$$

where $Q, K, V$ are all learnable parameters and $\sigma(\cdot)$ typically (but not always) represents the softmax non-linearity. In practice, one typically uses $m$ “heads” of attention which are concatenated along the feature dimension, and this is followed by a “channel-mixing” layer, or alternatively a classification head:

$$f_{MHSA}(X_i) := [f_1(X_i) \cdots f_m(X_i)] W$$

$$= \sum_{j=1}^m \sigma \left( \frac{X_i Q_j K_j^\top X_j}{\sqrt{d}} \right) X_j V_j W_j, \quad (10)$$

For the purpose of our analysis, noting that both $Q_j K_j^\top$ and $V_j W_j$ can be expressed by a single linear layer, we model the multi-head self-attention network as

$$f_{SA}(X_i) := \sum_{j=1}^m \frac{\sigma \left( X_i W_j L_j^\top \right)}{\sqrt{d}} X_j W_j, \quad (11)$$

We then define the multi-head self-attention training problem as follows

$$p_{SA}^* := \min_{W_{1j}, W_{2j}} \sum_{i=1}^n \mathcal{L}(f_{SA}(X_i), Y_i)$$

$$+ \frac{\beta}{2} \sum_{j=1}^m ||W_{1j}||^2_F + ||W_{2j}||^2_F. \quad (12)$$

We thus employ any generic convex loss function and standard weight-decay in our formulation. While direct convex analysis when $\sigma(\cdot)$ represents the softmax activation is intractable, we can analyze this architecture for many other activation functions. In particular, self-attention with both linear and ReLU activation functions has been proposed with performance on par to standard softmax activation networks (Shen et al., 2021; Yorsh et al., 2021; Yorsh & Kovalenko, 2021; Zhang et al., 2021). Thus, we will analyze linear, ReLU, and gated ReLU activation variants of the multi-head self-attention.

Theorem 3.1. For the linear activation multi-head self-attention training problem (12), for $m \geq m^*$ where $m^* \leq \min(d^2, d c)$, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p_{SA}^* = \min_{Z \in \mathbb{R}^{d^2 \times d c}} \sum_{i=1}^n \mathcal{L} \left( \sum_{k=1}^d \sum_{\ell=1}^d G_i[k, \ell] X_i Z^{(k, \ell)}, Y_i \right)$$

$$+ \frac{\beta}{2} ||Z||_2 \quad (13)$$

where $G_i := X_i^\top X_i$ and $Z^{(k, \ell)} \in \mathbb{R}^{d \times c}$.

Our result demonstrates that a linear activation self-attention model consists of a Gram (feature correlation) matrix weighted linear model, with a nuclear norm penalty which groups the individual models to each other.\textsuperscript{4}

One may also view the convex model as a set of linear models with a weighted nuclear norm, where each block $Z^{(k, \ell)}$ has a corresponding weight of $1/G_i[k, \ell]$. Thus, features with high correlation will have corresponding linear weights with larger norm. We note that when $\beta = 0$, the linear self-attention model (13) is equivalent to the linear two-layer MLP (6).

While typically the nuclear norm penalty on $Z$ has no corresponding norm on each individual linear model

\textsuperscript{4}Furthermore, there is a one-to-one mapping between the solutions of the convex and non-convex programs, which we describe in Appendix A.
While this linear model provides a simple, elegant explanation for the underpinnings of self-attention, we can also analyze self-attention blocks with non-linearities. We thus provide an analysis of ReLU-activation self-attention.

**Theorem 3.3.** For the ReLU activation multi-head self-attention training problem (12), define

$$X := \begin{bmatrix} X_1 \otimes X_1 & \cdots & X_n \otimes X_n \end{bmatrix}$$

and $$\{D_j\}_{j=1}^p := \{\text{diag}(I \{\{Xu_j \geq 0\}\}) : u_j \in \mathbb{R}^{d_p}\},$$

where $$P \leq 2r \left(\frac{t(n-1)}{r}\right)^r$$ and $$r := \text{rank}(X).$$ Then, for $$m \geq m^*$$ where $$m^* \leq ndc,$$ the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p_{SA}^* = \min_{Z \in \mathbb{R}^{d_p \times dc}} \frac{1}{n} \sum_{i=1}^n \left( \sum_{j=1}^p \sum_{k=1}^d \sum_{\ell=1}^d \mathbf{G}^{(k,\ell)}_{i,j} X_i Z_j^{(k,\ell)}, Y_i \right) + \beta \sum_{b=1}^B \|Z^{(b)}\|_*, \quad Z^{(b)} \in \mathbb{R}^{d_b \times d_b c}. \quad (14)$$

This corollary thus demonstrates that under the assumption of sets of uncorrelated features, a linear self-attention block separates over these sets. In particular, the blocks of $$Z$$ corresponding to values of 0 in the Gram matrix $$G_i$$ will be set to 0, eliminating interactions between uncorrelated features. This phenomenon is illustrated in Figure 1.

While this linear model provides a simple, elegant explanation for the underpinnings of self-attention, we can also analyze self-attention blocks with non-linearities. We thus provide an analysis of ReLU-activation self-attention.

Furthermore, while in the linear-activation case in (13), each linear model was scaled by a single entry in $$G_i,$$ in the ReLU case each linear model is scaled by a diagonal matrix $$G^{(k,\ell)}_{i,j}$$ which combines second-order information from $$X_i$$ with the hyperplane arrangements induced by the ReLU activation function. One may, for example, note the identity

$$G^{(k,\ell)}_{i,j} = \sum_{t=1}^s X_{i}[t,k]X_{i}[t,\ell]D^{(i,t)}_{j}, \quad (16)$$

for diagonal $$D^{(i,t)}_{j} \in \{0,1\}^{s \times s}.$$ Therefore, $$G^{(k,\ell)}_{i,j}$$ can be viewed as a correlation between features $$k$$ and $$\ell$$ weighted by diagonal $$\{0,1\}$$-valued hyperplane arrangements for each corresponding row $$t,$$ in other words a type of “local” correlation, where locality is achieved by the $$\{0,1\}$$ values in $$D^{(i,t)}_{j}.$$ This local correlation scales each token of the prediction, essentially giving weight to tokens which have been not been masked away by $$D^{(i,t)}_{j}.$$

### 4. Alternative Mixing Mechanisms

While self-attention is the original proposed token mixer used for vision transformers, there are many other alternative approaches which have shown to produce similar results while being more computationally efficient. We tackle two such architectures here.

#### 4.1. MLP Mixer

We begin by analyzing the MLP-Mixer architecture, an all-MLP alternative to self-attention networks with competitive performance on image classification benchmarks (Tolstikhin et al., 2021). The proposal is simple–apply an MLP along one dimension of the input, followed by an MLP along the opposite dimension. The simplest form of this MLP-Mixer architecture can thus be written as

$$p_{MM}^* := \min_{w_{1j}^i,w_{2j}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \sigma(w_{1j}^i X_i)w_{2j}, Y_i \right) + \frac{\beta}{2} \sum_{j=1}^m \|w_{1j}^i\|_F^2 + \|w_{2j}\|_F^2, \quad (17)$$

where $$\sigma$$ is an activation function. While (Tolstikhin et al., 2021) use the GeLU non-linearity (Hendrycks & Gimpel, 2016), we analyze the simpler linear and ReLU activation
counterparts, which shed important insights into the underlying structure of the MLP-Mixer architecture.

**Theorem 4.1.** For the linear activation MLP-Mixer training problem (17), for \( m \geq m^* \) where \( m^* = \min \{ s^2, dc \} \), the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[
p_{\text{MM}}^* = \min_{Z \in \mathbb{R}^{s \times dc}} \sum_{i=1}^{n} \mathcal{L} \left( \left[ f_1(X_i) \cdots f_c(X_i) \right], Y_i \right) + \beta \|Z\|_* = Z^p \text{vec}(X_i) \tag{18}
\]

where \( Z^p \in \mathbb{R}^{s \times d} \) for \( p \in [c] \), for \( Z^{(p,t)} \in \mathbb{R}^{s \times d} \) for \( t \in [s] \), and

\[
Z^p = \begin{bmatrix} Z^{(p,1)} & \cdots & Z^{(p,s)} \end{bmatrix}, Z = \begin{bmatrix} Z^{(1,1)} & \cdots & Z^{(c,1)} \\ \vdots & \ddots & \vdots \\ Z^{(1,s)} & \cdots & Z^{(c,s)} \end{bmatrix}
\]

We can contrast the fitting term of the linear MLP-Mixer to a standard linear MLP (6), where each column \( k \) of the network output is given by

\[
X_kZ^{(k)} = (Z^{(k)\top} \otimes I_s) \text{vec}(X_i),
\]

where \( Z^{(k)} \in \mathbb{R}^d \). Thus, the MLP-Mixer gives the network \( s^2 \) more degrees of freedom for fitting each column of \( Y_i \) than a standard linear MLP. This demonstrates that unlike the linear self-attention network, a linear MLP-Mixer model is not equivalent to a linear standard MLP even when \( \beta = 0 \). One may speculate that this additional implicit degrees of freedom allows mixer-like MLP models to fit complex distributions more easily compared to standard MLPs. While it appears from the fitting term that each output class of \( Y_i \) is fit independently, we note that these outputs are coupled together by the nuclear norm on \( Z \), which encourages \( \{ Z^{(k)} \}_{k=1}^c \) to be similar to one another.

Another interpretation of the convexified linear MLP-Mixer architecture may be achieved by simply permuting the columns of \( Z \) to form \( \tilde{Z} \), which does not affect the nuclear norm and thus does not impact the optimal solution. If one partitions the columns of \( \tilde{Z} \) according to blocks \( \tilde{Z}^{(t,k)} \in \mathbb{R}^{s \times c} \) for \( t \in [s], k \in [d] \), one can also write

\[
p_{\text{MM}}^* = \min_{\tilde{Z}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{t=1}^{s} \sum_{k=1}^{d} X_i[t,k] \tilde{Z}^{(t,k)}, Y_i \right) + \beta \|\tilde{Z}\|_* \tag{19}
\]

Here, the connections to the linear self-attention network (13) become more clear. While (13) is a weighted summation of linear models, where the weights correspond to Gram matrix entries, (19) is a weighted summation of predictions, where the weights correspond to data matrix entries. We also note that in most networks, typically \( s < d \), so the MLP-Mixer block has a lower order complexity to solve compared to the self-attention block. We can also extend these results to ReLU activation MLP-Mixers.

**Theorem 4.2.** For the ReLU activation MLP-Mixer training problem (17), we define

\[
X := \begin{bmatrix} X_1 \otimes I_s \\ \vdots \\ X_n \otimes I_s \end{bmatrix}, \quad \{ D_j \}_{j=1}^P := \{ \text{diag} (I \{ X_{uj} \geq 0 \}) : u_j \in \mathbb{R}^s \},
\]

where \( P \leq 2r \left( \frac{e(n-1)}{r} \right)^{r} \) and \( r := \text{rank}(X) \). Then, for \( m \geq m^* \) where \( m^* \leq ndc \), the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[
p_{\text{MM}}^* = \min_{Z_j} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{P} D_j Z_j^{(p,k)} \right) \text{vec}(X_i) + \beta \sum_{j=1}^{P} \| Z_j \|_{*,K_j} \tag{20}
\]

where

\[
f_p(X_i) := \sum_{j=1}^{P} D_j^{(i,1)} Z_j^{(p,1)} \cdots D_j^{(i,d)} Z_j^{(p,d)} \text{vec}(X_i)
\]

for \( D_j^{(i,k)} \in \mathbb{R}^{s \times s} \) and \( Z_j^{(p,k)} \in \mathbb{R}^{s \times s} \).

Now, unlike the self-attention model, where the effective data matrix for hyperplane arrangements was \( X_k \otimes X_i \), the MLP-mixer’s arrangements use \( X_k \otimes I_s \), providing additional degrees of freedom for partitioning the data while still incorporating only first-order information about the data. Using the same column permutation trick as in (19), one may write (20) as

\[
p_{\text{MM}}^* = \min_{\tilde{Z}_j} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{P} \sum_{t=1}^{s} \sum_{k=1}^{d} X_i[t,k] D_j^{(i,k)} \tilde{Z}_j^{(t,k)}, Y_i \right) + \beta \sum_{j=1}^{P} \| \tilde{Z}_j \|_{*,K_j}, \tag{21}
\]

where again now we see clearly the differences with ReLU self-attention (15), with the diagonal arrangements weighted by \( X_i \) rather than the Gram matrix, and instead of \( Z_j^{(t,k)} \) being weights of a linear model, they are simply predictions.

### 4.2. Fourier Neural Operator

In contrast to self-attention or MLP-like attention mechanisms, there is also a family of Fourier-based alternatives to
We note that FNO actually starkly resembles a two-layer CNN, where the first layer consists of a convolutional layer with full circulant padding and a global convolutional kernel. Unlike a typical CNN, in which the kernel size is usually small and the convolution is local, the convolution here is much larger, which means there are many more parameters than a typical CNN. Similar CNN architectures have previously been analyzed via convex duality (Ergen & Pi-Fanci, 2021a; Sahiner et al., 2021b). Accordingly, for both linear and ReLU activation, (24) is equivalent to a convex optimization problem.

**Theorem 4.4.** For the linear activation FNO training problem (24), for $m \geq m^*$ where $m^* \leq \min\{sd, c\}$, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p^*_{FN} = \min_{Z \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L}(\text{circ}(X_i)Z, Y_i) + \beta\|Z\|_*.$$  

**Theorem 4.5.** For the ReLU activation FNO training problem (24), we define

$$X := \begin{bmatrix} \text{circ}(X_1) \\ \vdots \\ \text{circ}(X_n) \end{bmatrix},$$

$$\{D_j\}_{j=1}^{P} := \{\text{diag}(1 \{Xu_j \geq 0\}) : u_j \in \mathbb{R}^{sd}\},$$

where $P \leq 2r \left(\frac{e(n-1)}{r}\right)^r$ and $r := \text{rank}(X)$. Then, for $m \geq m^*$ where $m^* \leq nc$, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p^*_{FN} = \min_{z_j \in \mathbb{R}^{sd}, \mathbf{K}_j} \sum_{i=1}^{n} \mathcal{L}(\sum_{j=1}^{P} D_j(i) \text{circ}(X_i)Z_{j}, Y_i) + \beta \sum_{j=1}^{P} \|Z_j\|_{*,\mathbf{K}_j}, \mathbf{K}_j := (2D_j - I_{ns})X.$$  

4.2.1. Block Diagonal FNO

While the FNO formulation is quite elegant, it requires many parameters ($d^2$ for each token). Accordingly, modifications have been proposed in the form of Adaptive Fourier Neural Operator (AFNO) (Guibas et al., 2021). One important modification pertains to enforcing the token weights to obey a block diagonal structure. This has been significantly improved the training and generalization ability of AFNO compared with the standard FNO. We call this architecture B-FNO, which boils down to

$$f_{B FN}(X_i) := \sigma\left(F^{-1} \begin{bmatrix} (FX_i)^{(1)} & \cdots & (FX_i)^{(s)} \end{bmatrix} W_1 \right)W_2.$$  

We note that FNO actually starkly resembles a two-layer CNN, where the first layer consists of a convolutional layer with full circulant padding and a global convolutional kernel. Unlike a typical CNN, in which the kernel size is usually small and the convolution is local, the convolution here is much larger, which means there are many more parameters than a typical CNN. Similar CNN architectures have previously been analyzed via convex duality (Ergen & Pi-Fanci, 2021a; Sahiner et al., 2021b). Accordingly, for both linear and ReLU activation, (24) is equivalent to a convex optimization problem.
Lemma 4.6. For weights $W_{1b} \in \mathbb{R}^{s \times d/B \times m/B}$ and $W_{2b} \in \mathbb{R}^{m/B \times c/B}$, assuming $\sigma$ operates element-wise, the B-FNO model (27) can be equivalently represented as

$$f_{B \text{FNO}}(X_i) = \left[ f^{(1)}_{B \text{FNO}}(X_i) \cdots f^{(B)}_{B \text{FNO}}(X_i) \right]$$

(28)

where $\text{circ}(X_i^{(b)}) \in \mathbb{R}^{s \times d/B}$ is a matrix composed of all $s$ circulant shifts of $X_i^{(b)} \in \mathbb{R}^{s \times d/B}$ along its first dimension.

Interestingly, the block-diagonal weights of AFNO contrast the local convolution in CNNs with a global and group-wise convolution with $B$ groups. We thus define

$$p_{BFN}^* := \min_{W_{1b}, W_{2b}} \sum_{i=1}^{n} \mathcal{L}(f_{B \text{FNO}}(X_i), Y_i)$$

(29)

$$+ \frac{\beta}{2} \sum_{b=1}^{B} \sum_{j=1}^{m} \|w_{1bj}\|^2 + \|w_{2bj}\|^2.$$  

Theorem 4.7. For the linear activation B-FNO training problem (29), for $m \geq m^*$ where $m^* \leq 1/B \min\{sd,c\}$, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p_{BFN}^* = \min_{Z_b} \sum_{i=1}^{n} \mathcal{L}\left( \left[ f^{(1)}(X_i) \cdots f^{(B)}(X_i) \right], Y_i \right),$$

$$+ \frac{\beta}{2} \sum_{b=1}^{B} \|Z_b\|_4,$$

(30)

$$Z_b \in \mathbb{R}^{sd/B \times c/B}, f^{(b)} := \text{circ}(X_i^{(b)})Z_b.$$  

Theorem 4.8. For the ReLU activation B-FNO training problem (29), we define

$$X_b := \left[ \text{circ}(X_i^{(1)}) \cdots \text{circ}(X_i^{(B)}) \right]$$

$$\{D_{b,j}\}_{j=1}^{P_b} := \{ \text{diag}(1_{X_b} u_j, 0) : u_j \in \mathbb{R}^{sd/B} \},$$

where $P_b \leq 2r_b (\frac{c(n)}{m})^{r_b}$ and $r_b := \text{rank}(X_b)$. Then, for $m \geq m^*$ where $m^* \leq nc/B$, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p_{BFN}^* = \min_{Z_{b,j}} \sum_{i=1}^{n} \mathcal{L}\left( \left[ f^{(1)}(X_i) \cdots f^{(B)}(X_i) \right], Y_i \right),$$

$$+ \frac{\beta}{2} \sum_{b=1}^{B} \sum_{j=1}^{P_b} \|Z_{j,b}\|_{\text{s,K},b,j},$$

(31)

where

$$Z_{b,j} \in \mathbb{R}^{sd/B \times c/B}, f^{(b)}(X_i) := \sum_{j=1}^{P_b} D_{b,j} \text{circ}(X_i^{(b)}) Z_{b,j}$$

$$K_{b,j} := (2D_{b,j} - I_n) X_b.$$  

5. Numerical Results

In this section, we seek to compare the performance of the transformer heads we have analyzed in this work to baseline convex optimization methods. This comparison allows us to illustrate the implicit biases imposed by these novel heads in a practical example. In particular, we consider the task of training a single new block of these convex heads for performing an image classification task. This is essentially the task of transfer learning without fine-tuning existing weights of the backbone network, which may be essential in computation and memory-constrained settings at the edge. For few-shot fine-tuning transformer tasks, non-convex optimization has observed to be unstable under different random initializations (Mosbach et al., 2020). Furthermore, fine-tuning only the final layer a network is a common practice, which performs very well in spurious correlation benchmarks (Kirichenko et al., 2022).

Specifically, we seek to classify images from the CIFAR-100 dataset (Krizhevsky et al., 2009). We first generate embeddings from a pretrained gMLP-S model (Liu et al., 2021) on $224 \times 224$ images from the ImageNet-1k dataset (Deng et al., 2009) with $16 \times 16$ patches ($s = 196, d = 256$). We then finetune the single convex head to classify images from CIFAR-100, while leaving the pre-trained backbone fixed.

For the backbone gMLP architecture, we reduce the feature dimension to $d = 100$ with average pooling as a pre-processing step before training with the convex heads. Similarly, for computational efficiency, we train the Gated ReLU variants of the standard ReLU architectures, since these Gated ReLU activation networks are unconstrained. For BFNO, we choose $B = 5$. All the heads use identical dimensions ($d = 100, s = 196, c = 100$), and we choose the number of neurons in ReLU heads to be $m = 100$, except in the case of self-attention, where we choose $m = 5$ to ensure the parameter count be roughly equal across heads (see Table 2 in Appendix B). As our baseline, we compare to a simple linear model (i.e., logistic regression) and to convex equivalents of MLPs as discussed in Section 2.1.

We summarize the results in Table 1. Here, we demonstrate that the attention variants outperform the standard convex MLP and linear baselines. This suggests that the higher-order information and additional degrees of freedom of the attention architectures provide advantages for difficult
Table 1. CIFAR-100 classification accuracy for training a single convex head. Embeddings are generated from gMLP-S pre-trained on ImageNet. Note that the backbone is not fine-tuned.

| Convex Head  | Act.  | Top-1  | Top-5  |
|--------------|-------|--------|--------|
| Self-Attention |      | 73.81  | 92.87  |
| MLP-Mixer    |      | **78.11** | **94.79** |
| B-FNO        | Linear | 68.68  | 90.62  |
| FNO          |      | 72.29  | 93.03  |
| MLP          |      | 65.95  | 89.33  |
| Linear       |      | 66.42  | 89.27  |
| Self-Attention |      | 74.74  | 93.45  |
| MLP-Mixer    |      | **80.22** | **95.79** |
| B-FNO        | Gated ReLU | 77.65  | 94.97  |
| FNO          |      | 72.93  | 92.71  |
| MLP          |      | 73.05  | 92.52  |

vision tasks. Surprisingly, for self-attention, FNO, and MLP-Mixer, there is only a marginal gap between the linear and ReLU activation performance, suggesting most of the benefit of these architectures is in their fundamental structure, rather than the nonlinearity which is applied. In contrast, for B-FNO, there is a very large gap between ReLU and linear activation accuracies, suggesting that this nonlinearity is more crucial when group convolutions are applied. These convexified architectures thus pave the way towards stable and transparent models for transfer learning.

6. Conclusion

We demonstrated that blocks of self-attention and common alternatives such as MLP-Mixer, FNO, and B-FNO are equivalent to convex optimization problems in the case of linear and ReLU activations. These equivalent convex formulations implicitly cluster correlated features, and are penalized with a block nuclear norm regularizer which ensures a global representation. For future work, it remains to craft efficient approximate solvers of these networks by leveraging the structure of these unique regularizers. Faster solvers may implemented, such as FISTA or related algorithms (some of which were explored in the context of convex MLPs (Mishkin et al., 2022)). In the long term for practical adoption, future theoretical work would also require analysis of deeper networks as are often used in practice. One may use this work in designing new network architectures by specifying the desired convex formulation.

ACKNOWLEDGMENTS

This work was partially supported by an Army Research Office Early Career Award, and the National Science Foundation under grants ECCS-2037304, DMS-2134248. Research reported in this work was supported by the National Institutes of Health under award numbers R01EB009690 and U01EB029427. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.
References

An, B., Lyu, J., Wang, Z., Li, C., Hu, C., Tan, F., Zhang, R., Hu, Y., and Chen, C. Repulsive attention: Rethinking multi-head attention as bayesian inference. *arXiv preprint arXiv:2009.09364*, 2020.

Bai, Y., Gautam, T., and Sojoudi, S. Efficient global optimization of two-layer relu networks: Quadratic-time algorithms and adversarial training. *arXiv preprint arXiv:2201.01965*, 2022.

Bartan, B. and Pilanci, M. Neural spectrahedra and semidefinite lifts: Global convex optimization of polynomial activation neural networks in fully polynomial-time. *arXiv preprint arXiv:2101.02429*, 2021a.

Bartan, B. and Pilanci, M. Training quantized neural networks to global optimality via semidefinite programming. In *International Conference on Machine Learning*, pp. 694–704. PMLR, 2021b.

Burer, S. and Monteiro, R. D. Local minima and convergence in low-rank semidefinite programming. *Mathematical programming*, 103(3):427–444, 2005.

Candès, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ergen, T. and Pilanci, M. Implicit convex regularizers of cnn architectures: Convex optimization of two-and three-layer networks in polynomial time. In *International Conference on Learning Representations (ICLR)* 2021, 2021a.

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

Ergen, T. and Pilanci, M. Global optimality beyond two layers: Training deep relu networks via convex programs. In *International Conference on Machine Learning*, pp. 2993–3003. PMLR, 2021c.

Ergen, T. and Pilanci, M. Revealing the structure of deep neural networks via convex duality. In *International Conference on Machine Learning*, pp. 3004–3014. PMLR, 2021d.

Ergen, T., Sahiner, A., Ozturkler, B., Pauly, J. M., Mardani, M., and Pilanci, M. Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization. In *International Conference on Learning Representations*, 2021.

Fiat, J., Malach, E., and Shalev-Shwartz, S. Decoupling gating from linearity. *arXiv preprint arXiv:1906.05032*, 2019.

Geng, Z., Guo, M.-H., Chen, H., Li, X., Wei, K., and Lin, Z. Is attention better than matrix decomposition? *arXiv preprint arXiv:2109.04553*, 2021.

Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., and Catanzaro, B. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Gupta, V., Bartan, B., Ergen, T., and Pilanci, M. Exact and relaxed convex formulations for shallow neural autoregressive models. In *International Conference on Acoustics, Speech, and Signal Processing*, 2021.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

Kim, H., Papamakarios, G., and Mnih, A. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pp. 5562–5571. PMLR, 2021.
Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *2009.*

Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2010.08895*, 2020.

Liu, H., Dai, Z., So, D. R., and Le, Q. V. Pay attention to mlps, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Magnus, J. R. and Neudecker, H. *Matrix differential calculus with applications in statistics and econometrics.* John Wiley & Sons, 2019.

Mardani, M., Mateos, G., and Giannakis, G. B. Decentralized sparsity-regularized rank minimization: Algorithms and applications. *IEEE Transactions on Signal Processing*, 61(21):5374–5388, 2013.

Mardani, M., Mateos, G., and Giannakis, G. B. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing*, 63(10):2663–2677, 2015.

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

Mosbach, M., Andriushchenko, M., and Klakow, D. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*, 2020.

Panahi, A., Saeedi, S., and Arodz, T. Shapeshifter: a parameter-efficient transformer using factorized reshaped matrices. *Advances in Neural Information Processing Systems*, 34, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pp. 7695–7705. PMLR, 2020.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

Rao, Y., Zhao, W., Zhu, Z., Lu, J., and Zhou, J. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34, 2021.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Sahiner, A., Ergen, T., Pauly, J. M., and Pilanci, M. Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. In *International Conference on Learning Representations*, 2020.

Sahiner, A., Ergen, T., Ozturkler, B., Bartan, B., Pauly, J., Mardani, M., and Pilanci, M. Hidden convexity of wasserstein gans: Interpretable generative models with closed-form solutions. *International Conference on Learning Representations*, 2021a.

Sahiner, A., Mardani, M., Ozturkler, B., Pilanci, M., and Pauly, J. M. Convex regularization behind neural reconstruction. In *ICLR*, 2021b.

Savarese, P., Evron, I., Soudry, D., and Srebro, N. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pp. 2667–2690. PMLR, 2019.

Shapiro, A. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009.

Shen, Z., Zhang, M., Zhao, H., Yi, S., and Li, H. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3531–3539, 2021.

Stanley, R. P. et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13(389-496):24, 2004.

Tatsunami, Y. and Taki, M. Raftmlp: Do mlp-based models dream of winning over computer vision? *arXiv preprint arXiv:2108.04384*, 2021.
Unraveling Attention via Convex Duality

Toh, K.-C. and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640): 15, 2010.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Izacard, G., Joulin, A., Synnaeve, G., Verbeek, J., et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, S., Li, B. Z., Khabisa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wang, Y., Lacotte, J., and Pilanci, M. The hidden convex optimization landscape of regularized two-layer relu networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*, 2021.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A nystrom-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021.

Yorsh, U. and Kovalenko, A. Pureformer: Do we even need attention? *arXiv preprint arXiv:2111.15588*, 2021.

Yorsh, U., Kordik, P., and Kovalenko, A. Simpletron: Eliminating softmax from attention computation. *arXiv preprint arXiv:2111.15588*, 2021.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S. Metaformer is actually what you need for vision. *arXiv preprint arXiv:2111.11418*, 2021.

Yun, C., Chang, Y.-W., Bhojanapalli, S., Rawat, A. S., Reddi, S. J., and Kumar, S. o(n) connections are expressive enough: Universal approximability of sparse transformers. *arXiv preprint arXiv:2006.04862*, 2020.

Zhang, B., Titov, I., and Sennrich, R. Sparse attention with linear units. *arXiv preprint arXiv:2104.07012*, 2021.
Appendix

A. Proofs

A.1. Preliminaries

Here, we describe the general technique which allows for convex duality of self-attention and similar architectures. In particular, all of the proofs for the theorems in the subsequent sections follow the following general form:

1. Re-scale the weights such that the optimization problem has a Frobenius norm penalty on the second layer weights with a norm constraint on the first layer weights.

2. Form the dual problem over the second-layer weights, creating a dual constraint which depends on the first layer weights.

3. Solve the dual constraint over the first layer weights.

4. Form the Lagrangian problem and solve over the dual weights.

5. Make any simplifications as necessary.

We describe the first two steps here, and use it in the following proofs.

Lemma A.1. Suppose we are given an optimization problem of the form

\[ p^* := \min_{\mathbf{w}_{1j}, \mathbf{w}_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} g(\mathbf{x}_i; \mathbf{w}_{1j}) \mathbf{w}_{2j}, \mathbf{y}_i \right) + \frac{\beta}{2} \sum_{j=1}^{m} \| \mathbf{w}_{1j} \|_F^2 + \| \mathbf{w}_{2j} \|_F^2, \] (32)

where \( g(\mathbf{x}_i; \mathbf{w}_{1j}) \) is any function such that \( g(\mathbf{x}_i; \mathbf{w}_{1j}) = \alpha_j g(\mathbf{x}_i; \mathbf{w}_{1j}) \) for \( \alpha_j > 0 \). Then, this problem is equivalent to

\[ p^* = \min_{\| \mathbf{w}_{1j} \|_F \leq 1, \mathbf{w}_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} g(\mathbf{x}_i; \mathbf{w}_{1j}) \mathbf{w}_{2j}, \mathbf{y}_i \right) + \beta \sum_{j=1}^{m} \| \mathbf{w}_{2j} \|_F. \] (33)

Proof. Due to the form of \( g(\mathbf{x}_i; \mathbf{w}_{1j}) \), we can rescale the parameters as \( \tilde{\mathbf{w}}_{1j} = \alpha_j \mathbf{w}_{1j}, \tilde{\mathbf{w}}_{2j} = \mathbf{w}_{2j}/\alpha_j \) for \( \alpha_j > 0 \) without changing the network output. Then, to minimize the regularization term, we can write the problem as

\[ p^* := \min_{\mathbf{w}_{1j}, \mathbf{w}_{2j}, \alpha_j > 0} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} g(\mathbf{x}_i; \mathbf{w}_{1j}) \mathbf{w}_{2j}, \mathbf{y}_i \right) + \frac{\beta}{2} \sum_{j=1}^{m} \alpha_j^2 \| \mathbf{w}_{1j} \|_F^2 + \| \mathbf{w}_{2j} \|_F^2/\alpha_j^2. \] (34)

Solving this minimization problem over \( \alpha_j \) (Savarese et al., 2019; Sahiner et al., 2020), we obtain

\[ p^* := \min_{\mathbf{w}_{1j}, \mathbf{w}_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} g(\mathbf{x}_i; \mathbf{w}_{1j}) \mathbf{w}_{2j}, \mathbf{y}_i \right) + \beta \sum_{j=1}^{m} \| \mathbf{w}_{1j} \|_F \| \mathbf{w}_{2j} \|_F. \] (35)

We can thus set \( \| \mathbf{w}_{1j} \|_F = 1 \) without loss of generality, and further relaxing this to \( \| \mathbf{w}_{1j} \|_F \leq 1 \) does not change the optimal solution. Thus, we are left with the desired result:

\[ p^* = \min_{\| \mathbf{w}_{1j} \|_F \leq 1, \mathbf{w}_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} g(\mathbf{x}_i; \mathbf{w}_{1j}) \mathbf{w}_{2j}, \mathbf{y}_i \right) + \beta \sum_{j=1}^{m} \| \mathbf{w}_{2j} \|_F. \] (36)

Note that the assumption on \( g \) encapsulates all architectures studied in this work: self-attention, MLP-Mixer, FNO, B-FNO, and other extensions with ReLU, gated ReLU, or linear activation functions all satisfy this property.
Lemma A.2. Suppose that

\[ p^* := \min_{\|W_1\|_F \leq 1, W_{2j}} \sum_{i=1}^{n} \mathcal{L}(R_i, Y_i) + \beta \sum_{j=1}^{m} \|W_{2j}\|_F. \]  

(37)

Then, for all \( \beta > 0 \), if \( m \geq m^* \) for some \( m^* \), this optimization problem is equivalent to

\[ p^* = \max_{\mathcal{V}_i} - \sum_{i=1}^{n} \mathcal{L}^*(\mathcal{V}_i, Y_i) \]

s.t. \( \max_{\|W_1\|_F \leq 1} \sum_{i=1}^{n} \mathcal{V}_i^T g(X_i; W_{1j}) \| F \leq \beta \),

(38)

where \( \mathcal{L}^* \) is the Fenchel conjugate of \( \mathcal{L} \).

Proof. We first can re-write the problem as

\[ p^* = \min_{\|W_1\|_F \leq 1} \min_{W_{2j}} \max_{\mathcal{V}_i} \sum_{i=1}^{n} \mathcal{L}(R_i, Y_i) + \beta \sum_{j=1}^{m} \|W_{2j}\|_F + \sum_{i=1}^{n} \text{trace}(\mathcal{V}_i^T R_i) - \sum_{i=1}^{n} \text{trace} \left( \mathcal{V}_i^T \sum_{j=1}^{m} g(X_i; W_{1j}) W_{2j} \right). \]  

(40)

By Sion’s minimax theorem, we can reverse the order of the outer maximum and minimum, and minimize this problem over \( W_{2j} \) and \( R_i \). Defining the Fenchel conjugate of \( \mathcal{L} \) as \( \mathcal{L}^*(\mathcal{V}_i, Y_i) := \max_{R_i} - \mathcal{L}(R_i, Y_i) + \text{trace}(\mathcal{V}_i^T R_i) \), we have

\[ p^* = \min_{\|W_1\|_F \leq 1} \max_{W_{2j}} \min_{\mathcal{V}_i} - \sum_{i=1}^{n} \mathcal{L}^*(\mathcal{V}_i, Y_i) + \beta \sum_{j=1}^{m} \|W_{2j}\|_F - \sum_{i=1}^{n} \text{trace} \left( \mathcal{V}_i^T \sum_{j=1}^{m} g(X_i; W_{1j}) W_{2j} \right). \]  

(41)

Now, we solve over \( W_{2j} \) to obtain

\[ p^* = \min_{\|W_1\|_F \leq 1} \max_{\mathcal{V}_i} - \sum_{i=1}^{n} \mathcal{L}^*(\mathcal{V}_i, Y_i) \]

(42)

Now, since Slater’s condition holds because \( \beta > 0 \), as long as \( m \geq m^* \) where \( m^* \) is the dimension of the constraints (see the individual cases for examples) we are permitted to switch the order of the maximum and minimum to obtain the desired result (Shapiro, 2009):

\[ p^* = \max_{\mathcal{V}_i} - \sum_{i=1}^{n} \mathcal{L}^*(\mathcal{V}_i, Y_i) \]

s.t. \( \max_{\|W_1\|_F \leq 1} \sum_{i=1}^{n} \mathcal{V}_i^T g(X_i; W_1) \| F \leq \beta \).

(43)

These two lemmas will prove invaluable in the subsequent proofs.
A.2. Proof of Theorem 3.1

We first note that for self-attention we remove the $1/\sqrt{d}$ factor for simplicity. We apply Lemmas A.1 and A.2 to (12) with the linear activation function to obtain

$$p_{SA}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t. } \max_{\|w_1\|_F \leq 1} \left\| \sum_{i=1}^{n} V_i^T (X_i W_i X_i^T) X_i \right\|_F \leq \beta.$$  \hspace{1cm} (44)

Due to the identity $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$ (Magnus & Neudecker, 2019), we can write this as

$$p_{SA}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t. } \max_{\|w_1\|_2 \leq 1} \left\| \sum_{i=1}^{n} ((X_i^T \otimes V_i^T)(X_i \otimes X_i)) w_1 \right\|_2 \leq \beta,$$  \hspace{1cm} (45)

where $w_1 = \text{vec}(W_1)$. We note that the norm constraint here has dimension $dc$ and $w_1$ has dimension $d^2$, so by (Shapiro, 2009) this strong duality result from Lemma A.2 requires that $m^* \leq \min\{d^2, dc\}$. We can further write this as

$$p_{SA}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t. } \max_{\|Z\|_* \leq 1} \text{trace} \left( \sum_{i=1}^{n} (X_i^T \otimes V_i^T)(X_i \otimes X_i) Z \right) \leq \beta.$$  \hspace{1cm} (46)

Now, we form the Lagrangian, given by

$$p_{SA}^* = \max_{\lambda \geq 0} \min_{\|Z\|_* \leq 1} \min_{\|v_i\|_2 \leq 1} \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \lambda \left( \beta - \sum_{i=1}^{n} \text{vec}((X_i \otimes X_i) Z) \right) \text{vec} \left( X_i \otimes V_i \right).$$ \hspace{1cm} (47)

By Sion’s minimax theorem, we are permitted to change the order of the maxima and minima, to obtain

$$p_{SA}^* = \min_{\lambda \geq 0} \max_{\|Z\|_* \leq 1} \min_{\|v_i\|_2 \leq 1} \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \lambda \left( \beta - \sum_{i=1}^{n} \text{vec}((X_i \otimes X_i) Z) \right) \text{vec} \left( X_i \otimes V_i \right).$$ \hspace{1cm} (48)

Now, defining $K_{c,s}$ as the $(c, s)$ commutation matrix we have the following identity (Magnus & Neudecker, 2019)

$$\text{vec}(X_i \otimes V_i) = ((I_d \otimes K_{c,s})(\text{vec}(X_i) \otimes I_c) \otimes I_s) \text{vec}(V_i).$$

Using this identity and maximizing over $V_i$, we obtain

$$p_{SA}^* = \min_{\|Z\|_* \leq 1} \min_{\lambda \geq 0} \sum_{i=1}^{n} \mathcal{L} \left( ((\text{vec}(X_i)^T \otimes I_c)(I_d \otimes K_{s,c} \otimes I_s) \text{vec}(X_i) \otimes X_i)(\lambda Z) \right), \text{vec}(Y_i) \right) + \beta \lambda.$$ \hspace{1cm} (49)

Rescaling such that $Z = \lambda Z$, we obtain

$$p_{SA}^* = \min_{\mathbf{Z} \in \mathbb{R}^{d \times dc}} \sum_{i=1}^{n} \mathcal{L} \left( ((\text{vec}(X_i)^T \otimes I_c)(I_d \otimes K_{s,c} \otimes I_s) \text{vec}(X_i) \otimes X_i) Z, \text{vec}(Y_i) \right) + \beta \|Z\|_*.$$ \hspace{1cm} (50)

It appears as though this is a very complicated function, but it actually simplifies greatly. In particular, one can write this as

$$p_{SA}^* = \min_{\mathbf{Z} \in \mathbb{R}^{d \times dc}} \sum_{i=1}^{n} \mathcal{L} \left( \mathbf{X}_i, \mathbf{Y}_i \right) + \beta \|Z\|_*$$

$$\mathbf{X}_i (o,p) := \sum_{k=1}^{d} \sum_{l=1}^{d} \sum_{t=1}^{s} X_i[t,l] X_i[t,k] X_i[o,l]^T Z[k,l].$$ \hspace{1cm} (51)
We suppose that \( G \) is block diagonal with \( B \) such blocks of size \( d_b \) such that \( \sum_b d_b = d \). Then, we have

\[
p^*_{SA} = \min_{Z \in \mathbb{R}^{d \times d_c}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{k=1}^{d} \sum_{\ell=1}^{d} G_i[k, \ell] X_i Z_i^{(k,\ell)} Y_i \right) + \beta \| Z \|_* \tag{53} \]

Lastly, we also demonstrate that there is a one-to-one mapping between the solution to (13) and (12). In particular, imagine we have a solution \( Z^* \) to (13) with optimal value \( p^*_{CVX} \). Let \( r := \text{rank}(Z^*) \), and take the SVD of \( Z^* \) as \( \sum_{j=1}^{r} \sigma_j u_j v_j^\top \), where \( u_j \in \mathbb{R}^{d^2} \) and \( v_j \in \mathbb{R}^{d_c} \). Let \( \text{vec}^{-1}(u_j) \in \mathbb{R}^{d^2} \) be the result of taking chunks of \( d \)-length vectors from \( u_j \) and stacking them in columns. Similarly, let \( \text{vec}^{-1}(v_j) \in \mathbb{R}^{d_c \times d} \) be the result of taking chunks of \( d \)-length vectors from \( v_j \) and stacking them in columns. Furthermore we will let \( \text{vec}^{-1}(u_j)_k \) be the \( k \)th column of \( \text{vec}^{-1}(u_j) \). Then, recognize that

\[
Z^{*(k,\ell)} = \sum_{j=1}^{r} \sigma_j \text{vec}^{-1}(u_j)_k \text{vec}^{-1}(v_j)^\top
\]

Thus, given \( Z^* \), we can form a candidate solution to (12) as follows:

\[
Z^* = \sum_{j=1}^{r} \sigma_j u_j v_j^\top
\]

\[
W_{1j} = \sqrt{\sigma_j} \text{vec}^{-1}(u_j)
\]

\[
W_{2j} = \sqrt{\sigma_j} \text{vec}^{-1}(v_j)^\top
\]

We then have

\[
p^*_{NCVX} = \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{r} X_i W_{1j} X_i^\top X_i W_{2j} Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{r} \| W_{1j} \|_F^2 + \| W_{2j} \|_F^2
\]

\[
= \sum_{i=1}^{n} \mathcal{L} \left( \sum_{k=1}^{d} \sum_{\ell=1}^{d} X_i \sum_{j=1}^{r} \sigma_j \text{vec}^{-1}(u_j)_k G_i[k, \ell] \text{vec}^{-1}(v_j)^\top Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{r} \| \sqrt{\sigma_j} u_j \|_2^2 + \| \sqrt{\sigma_j} v_j \|_2^2
\]

\[
= \sum_{i=1}^{n} \mathcal{L} \left( \sum_{k=1}^{d} \sum_{\ell=1}^{d} G_i[k, \ell] X_i Z_i^{*(k,\ell)} Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{r} \sigma_j + \sigma_j
\]

\[
= p^*_{CVX}
\]

Thus, the two solutions match. Similarly, if we have the solution \( (W_{1j}^*, W_{2j}^*) \) to (12), we can form the equivalent optimal convex weights to (13) as

\[
Z^* = \sum_{j=1}^{m} \text{vec}(W_{1j}^*) \text{vec}(W_{2j}^*)^\top
\]

and the same proof can be demonstrated in reverse.

A.3. Proof of Corollary 3.2

We suppose that \( G \) is block diagonal with \( B \) such blocks of size \( d_b \) such that \( \sum_b d_b = d \). Then, we have

\[
p^*_{SA} = \min_{Z \in \mathbb{R}^{d \times d_c}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{b=1}^{B} \sum_{j=1}^{d_b} \sum_{k=1}^{d_b} G_i^{(b)}[j, k] X_i Z_i^{(b,j,k)} Y_i \right) + \beta \| Z \|_* \tag{53} \]
Then, we have, for blocks $Z^{(b)} \in \mathbb{R}^{d_b \times d_b \times d_c}$,
\[
\|Z\|_* = \max_{\|A\|_2 \leq 1} \langle Z, A \rangle \geq \max_{\|A^{(b)}\|_2 \leq 1} \langle Z^{(b)}, A^{(b)} \rangle = \sum_{b=1}^{B} \|Z^{(b)}\|_* .
\] (54)

This lower bound for $Z$ is achievable without changing the fitting term, by letting
\[
Z = \begin{bmatrix}
Z^{(1)} & 0 & 0 \\
0 & Z^{(2)} & 0 \\
\vdots & & \ddots \\
0 & 0 & Z^{(B)}
\end{bmatrix}
\] (55)

Thus, in the block diagonal case, we have
\[
p^{*}_{SA} = \min_{\mathcal{Z}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{b=1}^{B} \sum_{k=1}^{d_b} \sum_{\ell=1}^{d_b} G^{(b)}_i[k, \ell] X_i Z^{(b,k,\ell)} Y_i \right) + \beta \sum_{b=1}^{B} \|Z^{(b)}\|_*
\] (56)
as desired.

\section*{A.4. Proof of Theorem 3.3}

We first note that for self-attention we remove the $1/\sqrt{d}$ factor for simplicity. We apply Lemmas A.1 and A.2 to (12) with the ReLU activation function to obtain
\[
p^{*}_{SA} = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^{N} V_i^\top (X_i \otimes X_i^\top) + X_i \right\|_F .
\] (57)

We again apply $\text{vec}(ABC) = (C^\top \otimes A) \text{vec}(B)$ (Magnus & Neudecker, 2019) to obtain
\[
p^{*}_{SA} = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|W_i\|_2 \leq 1} \left\| \sum_{i=1}^{N} (X_i \otimes V_i^\top)(X_i \otimes X_i)w_1 \right\|_2 .
\] (58)

Now, let $D^{(i)}_j \in \mathbb{R}^{s_2 \times s_2}$ be the $i$th block of $D_j$, and enumerate over all possible hyperplane arrangements $j$. Then, we have
\[
p^{*}_{SA} = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|W_i\|_2 \leq 1} \left\| \sum_{i=1}^{N} (X_i \otimes V_i^\top)D^{(i)}_j(X_i \otimes X_i)w_1 \right\|_2 .
\] (59)
We now can switch the order of max and min via Sion’s minimax theorem and maximize over \( dc \). We can also define sets \( C_j := \{ Z = u g^T \in \mathbb{R}^{d^2 \times dc} : K_j u \geq 0 \forall i, \| Z \|_* \leq 1 \} \). Then, we have

\[
p_{SA}^* = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i)
\]

\[
s.t. \quad \max_{j \in \{P\}} \quad \text{max} \\text{trace} \left( \sum_{i=1}^n (X_i^T \otimes V_i^T) D_j^{(i)} (X_i \otimes X_j) Z \right)
\]

Now, we simply need to form the Lagrangian and solve. The Lagrangian is given by

\[
p_{SA}^* = \max_{V_i} \min_{\lambda \geq 0} \min_{Z \in C_j} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) + \sum_{j=1}^P \lambda_j \left( \beta - \sum_{i=1}^n \text{vec} \left( D_j^{(i)} (X_i \otimes X_j) Z_j \right)^T \text{vec} (X_i \otimes V_i) \right)
\]

We now can switch the order of max and min via Sion’s minimax theorem and maximize over \( V_i \). Defining \( K_{c,s} \) as the \((c,s)\) commutation matrix \((\text{Magnus} & \text{Neudecker}, 2019)\):

\[
\text{vec}(X_i \otimes V_i) = (I_d \otimes K_{c,s})(\text{vec}(X_i) \otimes I_c) \otimes I_s) \text{vec}(V_i)
\]

Maximizing over \( V_i \), we have

\[
p_{SA}^* = \min_{\lambda \geq 0} \min_{Z_j \in C_j} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^P \left( (\text{vec}(X_i)^T \otimes I_c) (I_d \otimes K_{s,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)} (X_i \otimes X_j) \lambda_j Z_j \right), \text{vec}(Y_i) \right) + \beta \sum_{j=1}^m \lambda_j.
\]

Again rescaling \( \hat{Z}_j = \lambda_j Z_j \), we have

\[
p_{SA}^* = \min_{Z_j \in \mathbb{R}^{d^2 \times dc}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^P \left( (\text{vec}(X_i)^T \otimes I_c) (I_d \otimes K_{s,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)} (X_i \otimes X_j) Z_j \right), \text{vec}(Y_i) \right) + \beta \sum_{j=1}^m \| Z_j \|_{*,K_j}.
\]

It appears as though this is a very complicated function, but it actually simplifies greatly. In particular, one can write this as

\[
p_{SA}^* = \min_{Z_j \in \mathbb{R}^{d^2 \times dc}} \sum_{i=1}^n \mathcal{L} \left( \hat{Y}_i, Y_i \right) + \beta \| Z \|_{*,K_j}
\]

\[
\hat{Y}_i[a, p] := \sum_{j=1}^P \sum_{k=1}^d \sum_{l=1}^s X_i[t, l] X_i[t, k] D_j^{(t,m)} X_i[\ell, l]^T Z_j^{[k,l]}.
\]

Making any final simplifications, one obtains the desired result.

\[
p_{SA}^* = \min_{Z_j \in \mathbb{R}^{d^2 \times dc}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^P \sum_{k=1}^d \sum_{l=1}^s G_{i,j}^{(k,\ell)} X_j^{(k,\ell)} Y_i \right) + \beta \sum_{j=1}^m \| Z_j \|_{*,K_j}.
\]
Lastly, we also demonstrate that there is a one-to-one mapping between the solution to (15) and (12). In particular, imagine we have a solution \( \{Z_j^*_k\}_{j=1}^r \) to (15) with optimal value \( p_{CVX}^* \), where \( m^* \leq n \) are non-zero. Take the cone-constrained SVD of \( Z_j^*_k \) as \( \sum_{x=1}^{r_j} \sigma_{jx} u_{jx} v_{jx}^T \), where \( u_{jx}, v_{jx} \in \mathbb{R}^{d} \) and \( (2D_j - I_{ns^2})Xu_{jx} \geq 0 \), and \( v_{j} \in \mathbb{R}^{de} \). Let \( \text{vec}^{-1}(u_{jx}) \in \mathbb{R}^{d \times r} \) be the result of taking chunks of \( d \)-length vectors from \( u_{jx} \) and stacking them in columns. Similarly, let \( \text{vec}^{-1}(v_{j}) \in \mathbb{R}^{e \times d} \) be the result of taking chunks of \( e \)-length vectors from \( v_{j} \) and stacking them in columns. Furthermore we will let \( \text{vec}^{-1}(u_{j})_k \) be the \( k \)th column of \( \text{vec}^{-1}(u_{j}) \). Then, recognize that

\[
Z_j^{*(k,\ell)} = \sum_{x=1}^r \sigma_{jx} \text{vec}^{-1}(u_{jx}^k) \text{vec}^{-1}(v_{jx}^\ell)^T
\]

Thus, given \( Z^* \), we can form a candidate solution to (12) as follows:

\[
\hat{Z}_j^* = \sum_{x=1}^{r_j} \sigma_{jx} u_{jx} v_{jx}^T, \quad (2D_j - I_{ns^2})Xu_{jx} \geq 0, \quad \|u_{jx}\|_2 = 1, \quad \|v_{jx}\|_2 = 1
\]

\[
\hat{\mathbf{W}}_{1jx} = \sqrt{\sigma_{jx}} \text{vec}^{-1}(u_{jx})
\]

\[
\hat{\mathbf{W}}_{2jx} = \sqrt{\sigma_{jx}} \text{vec}^{-1}(v_{jx})^T
\]

We then have

\[
\hat{p}_{NCVX} = \sum_{i=1}^n \mathcal{L}(\sum_{j=1}^m \sum_{x=1}^{r_j} \sigma_{jx} (\mathbf{X}_i \mathbf{W}_{1jx} \mathbf{X}_i^T) + \mathbf{X}_i \mathbf{W}_{2jx}, Y_i) + \frac{\beta}{2} \sum_{j=1}^m \sum_{x=1}^{r_j} \|\mathbf{W}_{1jx}\|_F^2 + \|\mathbf{W}_{2jx}\|_F^2
\]

\[
= \sum_{i=1}^n \mathcal{L}(\sum_{j=1}^m \sum_{x=1}^{r_j} \sigma_{jx} \left( \text{diag}^{-1}(D_j^{(\ell)}) \odot (\text{vec}^{-1}(u_{jx}) \mathbf{X}_i^T) \right) \mathbf{X}_i \text{vec}^{-1}(v_{jx})^T, Y_i)
\]

\[
+ \frac{\beta}{2} \sum_{j=1}^m \sum_{x=1}^{r_j} \|\sqrt{\sigma_{jx}} u_{jx}\|_2^2 + \|\sqrt{\sigma_{jx}} v_{jx}\|_2^2
\]

\[
= \sum_{i=1}^n \mathcal{L}(\sum_{j=1}^m \sum_{x=1}^{r_j} \sigma_{jx} \mathbf{G}_{i,j}^{(k,\ell)} \mathbf{X}_i \text{vec}^{-1}(u_{jx}) \text{vec}^{-1}(v_{jx})^T, Y_i) + \frac{\beta}{2} \sum_{j=1}^m \sum_{x=1}^{r_j} \sigma_{jx} + \sigma_{jx}
\]

\[
= \sum_{i=1}^n \mathcal{L}(\sum_{j=1}^m \sum_{x=1}^{r_j} \mathbf{G}_{i,j}^{(k,\ell)} \mathbf{X}_i Z_j^{*(k,\ell)}, Y_i)
\]

\[
+ \beta \sum_{j=1}^m \|Z_j^*\|_{*,K_j}
\]

\[
= p_{CVX}^*
\]

Thus, the two solutions match.

\[\square\]

\textbf{A.5. Proof of Theorem 4.1}

We apply Lemmas A.1 and A.2 to (17) with the linear activation function to obtain

\[
p_{MM}^* = \max_{\mathbf{V}_i} - \sum_{i=1}^n \mathcal{L}^*(\mathbf{V}_i, Y_i)
\]

s.t. \( \max_{\mathbf{W}_i} \|\sum_{i=1}^n \mathbf{V}_i^T \mathbf{W}_i \|_F \leq \beta \) \quad (67)

We again apply \( \text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \) (Magnus & Neudecker, 2019) and maximize over \( \text{vec}(\mathbf{W}_1) \) to obtain

\[
p_{MM}^* = - \max_{\mathbf{V}_i} \sum_{i=1}^n \mathcal{L}^*(\mathbf{V}_i, Y_i)
\]

s.t. \( \|\sum_{i=1}^n \mathbf{X}_i^T \otimes \mathbf{V}_i\|_2 \leq \beta. \)
We note that the norm constraint here has dimension $dc$ and $\text{vec}(W_1)$ has dimension $s^2$, so by (Shapiro, 2009) this strong duality result from Lemma A.2 requires that $m^* \leq \min\{s^2, dc\}$. The Lagrangian is given by

$$p_{MM}^* = \max_{\lambda \geq 0} \min_{\|Z\|_1 \leq 1} \left( -\sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^{m} \lambda_j \left( \beta - \sum_{i=1}^{n} \text{trace} \left( (I_d \otimes V_i^T)(X_i^T \otimes I_s)Z \right) \right) \right)$$

which by Sion’s minimax theorem and simplification can also be written as

$$p_{MM}^* = \min_{\lambda \geq 0} \min_{\|Z\|_1 \leq 1} \max_{V_i} \left( -\sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^{m} \lambda_j \left( \beta - \sum_{i=1}^{n} \text{vec}(X_i^T \otimes I_s)Z)^T \text{vec}(I_d \otimes V_i) \right) \right)$$

We define $K_{c,d}$ as the $(c, d)$ commutation matrix (Magnus & Neudecker, 2019):

$$\text{vec}(I_d \otimes V_i) = (I_d \otimes K_{c,d})(\text{vec}(I_d) \otimes I_c) \text{vec}(V_i)$$

Maximizing over $V_i$, followed by re-scaling $\hat{Z} = \lambda Z$ gives us

$$p_{MM}^* = \min_{Z \in \mathbb{R}^{s \times dc}} \sum_{i=1}^{n} \mathcal{L} \left( (\text{vec}(I_d)^T \otimes I_c)(I_d \otimes K_{d,c}) \otimes I_s) \text{vec}(X_i) , \text{vec}(Y_i) + \beta \|Z\|_\star \right)$$

Making any final simplifications, one obtains the desired result.

$$p_{MM}^* = \min_{Z \in \mathbb{R}^{s \times dc}} \sum_{i=1}^{n} \mathcal{L} \left( [f_1(X_i) \cdots f_c(X_i)] , Y_i \right) + \beta \|Z\|_\star$$

Lastly, we also demonstrate that there is a one-to-one mapping between the solution to (18) and (17). In particular, we have a solution $\hat{Z}^*$ to (18) with optimal value $p_{C_Y}^*$. First, rearrange the solution to be of the form $\hat{Z}^*$ of (19). Let $r := \text{rank}(\hat{Z}^*)$, and take the SVD of $\hat{Z}^*$ as $\sum_{j=1}^{r} \sigma_j u_j v_j^T$, where $u_j \in \mathbb{R}^{s^2}$ and $v_j \in \mathbb{R}^{dc}$. Let $\text{vec}^{-1}(u_j) \in \mathbb{R}^{s \times s}$ be the result of taking chunks of $s$-length vectors from $u_j$ and stacking them in columns. Similarly, let $\text{vec}^{-1}(v_j) \in \mathbb{R}^{c \times d}$ be the result of taking chunks of $c$-length vectors from $v_j$ and stacking them in columns. Furthermore we will let $\text{vec}^{-1}(u_j)_t$ be the $t$th column of $\text{vec}^{-1}(u_j)$. Then, recognize that

$$\hat{Z}^{*(i,k)} = \sum_{j=1}^{r} \sigma_j \text{vec}^{-1}(u_j)_i \text{vec}^{-1}(v_j)_k^T$$

Thus, given $\hat{Z}^*$, we can form a candidate solution to (12) as follows:

$$\hat{Z}^* = \sum_{j=1}^{r} \sigma_j u_j v_j^T$$

$$\hat{W}_{1j} = \sqrt{\sigma_j} \text{vec}^{-1}(u_j)$$

$$\hat{W}_{2j} = \sqrt{\sigma_j} \text{vec}^{-1}(v_j)^T$$
We apply Lemmas A.1 and A.2 to (17) with the ReLU activation function to obtain

\[ p_{NCVX} = \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{r} \tilde{W}_{1j} X_i, \tilde{W}_{2j}, Y_i \right) + \beta \sum_{j=1}^{r} \| \tilde{W}_{1j} \|_F^2 + \| \tilde{W}_{2j} \|_F^2. \]

We note that the norm constraint here has dimension \( dc \).

Thus, the two solutions match.

A.6. Proof of Theorem 4.2

We apply Lemmas A.1 and A.2 to (17) with the ReLU activation function to obtain

\[ p_{MM}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \]

subject to

\[ \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^{n} V_i^T (W_i X_i)_+ \right\|_F \leq \beta \]

This is equivalent to (Magnus & Neudecker, 2019)

\[ p_{MM}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \]

subject to

\[ \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^{n} (I_d \otimes V_i^T) ((X_i^T \otimes I_d) \text{vec}(W_i))_+ \right\|_F \leq \beta \]

We note that the norm constraint here has dimension \( dc \), so by (Shapiro, 2009; Pilanci & Ergen, 2020) this strong duality result from Lemma A.2 requires that \( m^* \leq ndc \). Now, let \( D_j^{(i)} \in \mathbb{R}^{ad \times sd} \) be the \( i \)th block of \( D_j \), and enumerate over all possible hyperplane arrangements \( j \). Then, we have

\[ p_{MM}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \]

subject to

\[ \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^{n} (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_d) w_1 \right\|_2 \leq \beta. \]

Now, using the concept of dual norm, this is equal to

\[ p_{MM}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \]

subject to

\[ \max_{\|g\| \leq 1, \|w_1\|_2 \leq 1} \sum_{i=1}^{n} (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_d) w_1 \leq \beta \]
We can also define sets $C_j := \{ Z = u^T \in \mathbb{R}^{s \times dc} : K_j u \geq 0, \| Z \|_* \leq 1 \}$. Then, we have

\[
p_{MM}^* = \max_{V_i} - \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) \\
\text{s.t.} \quad \max_{\lambda \geq 0, Z \in C_j} \sum_{i=1}^{P} \left( \sum_{j=1}^{n} (I_d \otimes V_i^T)D_j^{(i)}(X_i^T \otimes I_s)Z \right) \leq \beta
\]  

(76)

Now, we simply need to form the Lagrangian and solve. The Lagrangian is given by

\[
p_{MM}^* = \max_{V_i} \min_{\lambda \geq 0} \min_{Z \in C_j} \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) + \sum_{j=1}^{P} \lambda_j \left( \beta - \sum_{i=1}^{n} \text{vec} \left( (I_d \otimes V_i^T)D_j^{(i)}(X_i^T \otimes I_s)Z \right) \right)
\]  

(77)

We now can switch the order of max and min via Sion’s minimax theorem and maximize over $V_i$:

\[
p_{MM}^* = \min_{\lambda \geq 0} \min_{Z \in C_j} \sum_{i=1}^{n} \mathcal{L}^* (V_i, Y_i) + \sum_{j=1}^{P} \lambda_j \left( \beta - \sum_{i=1}^{n} \text{vec} \left( D_j^{(i)}(X_i^T \otimes I_s)Z_j \right)^T \text{vec} (I_d \otimes V_i) \right)
\]  

(78)

Now, defining $K_{c,d}$ as the $(c,d)$ commutation matrix:

\[
\text{vec}(I_d \otimes V_i) = ((I_d \otimes K_{c,d})(\text{vec}(I_d) \otimes I_c) \otimes I_s) \text{vec}(V_i)
\]

Solving over $V_i$ yields

\[
p_{MM}^* = \min_{\lambda \geq 0} \sum_{j=1}^{P} \left( \sum_{i=1}^{n} (\text{vec}(I_d)^T \otimes I_c)(I_d \otimes K_{d,c}) \otimes I_s) \text{vec} \left( D_j^{(i)}(X_i^T \otimes I_s)Z_j \right) \right) + \beta \sum_{j=1}^{P} \lambda_j
\]  

(79)

Re-scaling $\tilde{Z}_j = \lambda_j Z_j$ gives us

\[
p_{MM}^* = \min_{\tilde{Z}_j} \sum_{i=1}^{n} \left( \sum_{j=1}^{P} (\text{vec}(I_d)^T \otimes I_c)(I_d \otimes K_{d,c}) \otimes I_s) \text{vec} \left( D_j^{(i)}(X_i^T \otimes I_s)Z_j \right) \right) + \beta \sum_{j=1}^{P} \| Z_j \|_{*,K_j}
\]  

(80)

One can actually greatly simplify this result, and can re-write this as

\[
p_{MM}^* = \min_{Z_j \in \mathbb{R}^{s \times dc}} \sum_{i=1}^{n} \left( \frac{1}{n} \sum_{j=1}^{P} \left( f_1(X_i) \cdots f_c(X_i) \right), Y_i \right) + \beta \sum_{j=1}^{P} \| Z_j \|_{*,K_j}
\]  

(81)

\[
f_p(X_i) := \sum_{j=1}^{P} \left[ D_j^{(i,1)}Z_j^{(p,1)} \cdots D_j^{(i,d)}Z_j^{(p,d)} \right] \text{vec}(X_i).
\]  

(82)

Lastly, we also demonstrate that there is a one-to-one mapping between the solution to (20) and (17). In particular, we have a solution $\{ Z_j^{(i)} \}_{j=1}^{m_i}$ to (20) with optimal value $p_{CV,X}^*$, where $m_i \leq n$ are non-zero. First, rearrange the solution to be of the form $\tilde{Z}_j$ of (21). Take the cone-constrained SVD of $\tilde{Z}_j$ as $z_{jx} u_{jx} v_{jx}^T$, where $u_{jx} \in \mathbb{R}^s$ and $v_{jx} \in \mathbb{R}^{dc}$. Let vec$^{-1}(u_{jx}) \in \mathbb{R}^{s \times s}$ be the result of taking chunks of $s$-length vectors from $u_{jx}$ and stacking them in columns. Similarly, let vec$^{-1}(v_{jx}) \in \mathbb{R}^{c \times d}$ be the result of taking chunks of $c$-length vectors from $v_{jx}$ and stacking them in columns. Furthermore, we will let vec$^{-1}(u_{jx})_t$ be the $t$th column of vec$^{-1}(u_{jx})$. Then, recognize that

\[
\tilde{Z}_j^{(i,k)} = \sum_{x=1}^{r_j} \sigma_{jx} \text{vec}^{-1}(u_{jx})_t \text{vec}^{-1}(v_{jx})_k^T
\]
We then have

\[
\tilde{Z}_j^* = \sum_{x=1}^{\tau} \sigma_{jx} u_{jx} v_{jx}^\top, \quad (2D_j - I_{nd}) X u_{jx} \geq 0, \quad \|u_{jx}\|_2 = 1, \quad \|v_{jx}\|_2 = 1
\]

\[
\hat{W}_{1j} = \sqrt{\sigma_j} \text{vec}^{-1}(u_{jx})
\]

\[
\hat{W}_{2j} = \sqrt{\sigma_j} \text{vec}^{-1}(v_{jx})^\top
\]

We then have

\[
\hat{p}_{NCVX} = \sum_{i=1}^{n} L \left( \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} (\hat{W}_{1j} X_i)_x + \hat{W}_{2j}, Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} \|\hat{W}_{1j}\|_F^2 + \|\hat{W}_{2j}\|_F^2
\]

\[
= \sum_{i=1}^{n} L \left( \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} \sigma_j(\text{vec}^{-1}(u_{jx}) X_i)_x + \text{vec}^{-1}(v_{jx})^\top, Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} \|\sqrt{\sigma_j} u_{jx}\|_2^2 + \|\sqrt{\sigma_j} v_{jx}\|_2^2
\]

\[
= \sum_{i=1}^{n} L \left( \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} \sigma_j(\text{vec}^{-1}(u_{jx}) X_i)_x + \text{vec}^{-1}(v_{jx})^\top, Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} \sigma_j + \sigma_j
\]

\[
= \sum_{i=1}^{n} L \left( \sum_{j=1}^{m^*} \sum_{x=1}^{r_j} X_i[k, \ell] D_j^{(i,k)} \tilde{Z}_j^{(i,k), x} Y_i \right) + \beta \sum_{j=1}^{m^*} \|\tilde{Z}_j^*\|_*
\]

Thus, the two solutions match. \(\square\)

**A.7. Proof of Lemma 4.3**

The expression (22) can equivalently be written as

\[
f_{FN}(X_i) = \sigma \left( F^{-1} \left[ F_j^\top X_i \right] \cdots \left[ \mathbf{V}^{(1)} \right] \left[ \mathbf{W}_1 \right] \right) \mathbf{W}_2
\]

and further as

\[
f_{FN}(X_i) = \sigma \left( F^{-1} \left[ F_j^\top X_i \right] \cdots \left( F \otimes I_d \left[ L^{(1)} \right] \cdots \left[ L^{(s)} \right] \right) \mathbf{W}_1 \right) \mathbf{W}_2,
\]

which simplifies to

\[
f_{FN}(X_i) = \sigma \left( \left[ X_i \quad X_{i(1)} \cdots X_{i(s)} \right] L \right) \mathbf{W}_1 \mathbf{W}_2,
\]

where \(X_{i(u)}\) is \(X_i\) circularly shifted by \(u\) spots in its first dimension, and \(K\) has been reshaped to the form \(\mathbb{R}^{s \times d \times d}\). Noting that we can merge \(L \mathbf{W}_1\) into one matrix without losing any expressibility, we obtain

\[
f_{FN}(X_i) = \sigma \left( \text{circ}(X_i) \mathbf{W}_1 \right) \mathbf{W}_2 = \sum_{j=1}^{m} \sigma \left( \text{circ}(X_i) w_{1j} \right) w_{2j}^\top
\]

as desired. \(\square\)
A.8. Proof of Theorem 4.4

We start with the problem

\[ p^*_{FN} = \min_{w_{1j}, w_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} \text{circ}(X_i) w_{1j} w_{2j}^T, Y_i \right) + \frac{\beta}{2} \sum_{j=1}^{m} \|w_{1j}\|_2^2 + \|w_{2j}\|_2^2 \]  \tag{87}

We now apply Lemmas A.1 and A.2 to obtain

\[ p^*_{FN} = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

s.t. \[ \max_{\|w_1\|_2 \leq 1} \| \sum_{i=1}^{n} V_i^T \text{circ}(X_i) w_1 \|_2 \leq \beta \]  \tag{88}

We note that the norm constraint here has dimension \( c \) and \( w_1 \) has dimension \( sd \), so by (Shapiro, 2009) this strong duality result from Lemma A.2 requires that \( m^* \leq \min\{sd, c\} \). This is equivalent to

\[ p^*_{FN} = \max_{V_i} \min_{\lambda \geq 0, \|Z\|_1 \leq 1} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \lambda (\beta - \text{trace}(Z^T \sum_{i=1}^{n} \text{circ}(X_i)^T V_i)). \]  \tag{90}

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( V_i \)

\[ p^*_{FN} = \min_{\lambda \geq 0, \|Z\|_1 \leq 1} \sum_{i=1}^{n} \mathcal{L}(\text{circ}(X_i) Z, Y_i) + \beta \lambda. \]  \tag{91}

Lastly, we rescale \( \tilde{Z} = \lambda Z \) to obtain

\[ p^*_{FN} = \min_{Z \in \mathbb{R}^{sd \times c}} \sum_{i=1}^{n} \mathcal{L}(\text{circ}(X_i) Z, Y_i) + \beta \|Z\|_1. \]  \tag{92}

as desired. \( \square \)

A.9. Proof of Theorem 4.5

We now apply Lemmas A.1 and A.2 with the ReLU activation function to obtain

\[ p^*_{FN} = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

s.t. \[ \max_{\|w_1\|_2 \leq 1} \| \sum_{i=1}^{n} V_i^T \text{circ}(X_i) w_1 \|_2 \leq \beta \]  \tag{93}

We note that the norm constraint here has dimension \( c \), so by (Shapiro, 2009; Pilanci & Ergen, 2020) this strong duality result from Lemma A.2 requires that \( m^* \leq nc \). We introduce hyperplane arrangements \( D_j \) and enumerate over all of them,
yielding
\[
p_{FN}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
\]
\[
s.t. \max_{\|w_i\|_2 \leq 1 \atop j \in [P] \atop K_j, w_i \geq 0} \sum_{i=1}^{n} V_i^T D_j^{(i)} \circ(X_i) w_i \leq \beta.
\]

Using the concept of dual norm, this is equivalent to
\[
p_{FN}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
\]
\[
s.t. \max_{\|g\|_2 \leq 1 \atop \|w_i\|_2 \leq 1 \atop j \in [P] \atop K_j, w_i \geq 0} \sum_{i=1}^{n} V_i^T D_j^{(i)} \circ(X_i) w_i \leq \beta.
\]

We can also define sets \( C_j := \{Z = u g^T \in \mathbb{R}^{2 \times dc} : K_j u \geq 0, \|Z\|_* \leq 1\} \). Then, we have
\[
p_{FN}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
\]
\[
s.t. \max_{j \in [P] \atop Z \in C_j} \text{trace} \left( \sum_{i=1}^{n} V_i^T D_j^{(i)} \circ(X_i) Z \right) \leq \beta.
\]

We form the Lagrangian as
\[
p_{FN}^* = \max_{V_i} \min_{\lambda \geq 0} \min_{Z_j \in C_j} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^{P} \lambda_j (\beta - \text{trace}(Z_j^* \sum_{i=1}^{n} D_j^{(i)} \circ(X_j) V_i)).
\]

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( V_i \)
\[
p_{FN}^* = \min_{\lambda \geq 0} \min_{Z_j \in C_j} \left( \sum_{i=1}^{n} \mathcal{L}(\sum_{j=1}^{P} D_j^{(i)} \circ(X_j) Z_j, Y_i) + \beta \sum_{j=1}^{P} \lambda_j \right).
\]

Lastly, we rescale \( Z_j = \lambda_j Z_j \) to obtain
\[
p_{FN}^* = \min_{Z_j \in \mathbb{R}^{dx \times e}} \left( \sum_{i=1}^{n} \mathcal{L}(\sum_{j=1}^{P} D_j^{(i)} \circ(X_j) Z_j, Y_i) + \beta \sum_{j=1}^{P} \|Z_j\|_{*, K_j} \right).
\]

as desired.

**A.10. Proof of Lemma 4.6**

The expression (28) can equivalently be written as
\[
f_{BFN}(X_i) = \sigma \left( F^{-1} \begin{bmatrix} F_i^\top X_i \\ & \ddots \\ & & F_i^\top X_i \end{bmatrix} \begin{bmatrix} V^{(1)} \\ \vdots \\ V^{(s)} \end{bmatrix} \right) W_2
\]

and further as
\[
f_{BFN}(X_i) = \sigma \left( F^{-1} \begin{bmatrix} F_i^\top X_i \\ & \ddots \\ & & F_i^\top X_i \end{bmatrix} \left( F \otimes I_d \begin{bmatrix} L^{(1)} \\ \vdots \\ L^{(s)} \end{bmatrix} \right) \right) W_2,
\]
which simplifies to

$$f_{BFN}(X_i) = \sigma \left( \left[ \text{circ}(X_i^{(1)}) \cdots \text{circ}(X_i^{(B)}) \right] L \right) W_2,$$

(102)

where $X_{i(u)}$ is $X_i$ circularly shifted by $u$ spots in its first dimension, and $L$ has been reshaped to the form $\mathbb{R}^{sd \times d}$. Without loss of generality, noting the block diagonal structure over blocks of $L$, we permute the rows and corresponding columns of $L$ so that $L$ is a block diagonal matrix

$$f_{BFN}(X_i) = \sigma \left( \left[ \text{circ}(X_i^{(1)}) \cdots \text{circ}(X_i^{(s)}) \right] L \right) W_2.$$

(103)

Simplifying the term inside, we have

$$f_{BFN}(X_i) = \sigma \left( \left[ \text{circ}(X_i^{(1)l}) \cdots \text{circ}(X_i^{(B)}) L \right] \right) W_2.$$

(104)

Now, assuming $\sigma$ is applied elementwise, we have

$$f_{BFN}(X_i) = \sigma \left( \left[ \text{circ}(X_i^{(1)l}) \cdots \text{circ}(X_i^{(B)}) L \right] \right) W_2.$$

(105)

Lastly, we use the block structure on $W_2$ to obtain

$$f_{BFN}(X_i) = \sigma \left( \left[ \text{circ}(X_i^{(1)l}) W_2^{(1)} \cdots \text{circ}(X_i^{(B)}) W_2^{(B)} \right] \right)$$

which we can write equivalently as

$$f_{BFN}(X_i) = \left[ f_{BFN}^{(1)}(X_i) \cdots f_{BFN}^{(B)}(X_i) \right]$$

(106)

(107)

as desired.

\[\square\]

A.11. Proof of Theorem 4.7

We now apply Lemmas A.1 and A.2 with ReLU activation obtain

$$p^*_B N = \max_{V_i} \min_{\lambda \geq 0} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t. } \max_{\|w_{1b}\|_2 \leq 1} \| \sum_{i=1}^{n} \text{circ}(X_i^{(b)}) w_{1b} \|_2 \leq \beta \forall b \in [B]$$

(108)

We note that the norm constraint here has dimension $c/B$ and $w_{1b}$ has dimension $sd/B$, so by (Shapiro, 2009) this strong duality result from Lemma A.2 requires that $m^* \leq 1/B \min \{sd, c\}$. This is equivalent to

$$p^*_B N = \max_{V_i} \min_{\lambda \geq 0} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t. } \| \sum_{i=1}^{n} \text{circ}(X_i^{(b)}) \|_2 \leq \beta \forall b \in [B]$$

(109)

We form the Lagrangian as

$$p^*_B N = \max_{V_i} \min_{\lambda \geq 0} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{b=1}^{B} \lambda_b (\beta - \text{trace}(Z_{ib} \sum_{i=1}^{n} \text{circ}(X_i^{(b)})^{\top} V_i^{(b)})).$$

(110)
We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over $V_i^{(b)}$

$$p_{BFN}^* = \min_{\lambda \geq 0} \min_{\|Z_b\|_* \leq 1} \sum_{i=1}^n \mathcal{L}(\left[ \lambda_1 \circ \text{circ}(X_i^{(1)}), \ldots, \lambda_B \circ \text{circ}(X_i^{(B)}) \right], Y_i) + \beta \sum_{j=1}^B \lambda_b. \tag{111}$$

Lastly, we rescale $\tilde{Z}_b = \lambda_b Z_b$ to obtain

$$p_{BFN}^* = \min_{Z \in \mathbb{R}^{d \times c}} \sum_{i=1}^n \mathcal{L}(\left[ \text{circ}(X_i^{(1)}), \ldots, \text{circ}(X_i^{(B)}) \right], Y_i) + \beta \|Z\|_* \tag{112}$$
as desired.

**A.12. Proof of Theorem 4.8**

We now apply Lemmas A.1 and A.2 with the ReLU activation function to obtain

$$p_{BFN}^* = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \tag{113}$$
s.t. $\max_{\|w_{1b}\|_2 \leq 1} \| \sum_{i=1}^n V_i^{(b)}(\circ \text{circ}(X_i^{(b)}))w_{1b}\|_2 \leq \beta \forall b \in [B]$

We note that the norm constraint here has dimension $c/B$, so by (Shapiro, 2009; Pilanci & Ergen, 2020) this strong duality result from Lemma A.2 requires that $n^* \leq nc/B$. We introduce hyperplane arrangements $D_{b,j}$ and enumerate over all of them, yielding

$$p_{BFN}^* = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \tag{114}$$
s.t. $\max_{\|w_{1b}\|_2 \leq 1} \| \sum_{i=1}^n V_i^{(b)}(D_{b,j})\circ \text{circ}(X_i^{(b)})w_{1b}\|_2 \leq \beta \forall b \in [B]$. 

Using the concept of dual norm, this is equivalent to

$$p_{BFN}^* = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \tag{115}$$
s.t. $\max_{\|g_1\|_2 \leq 1} \| \sum_{i=1}^n V_i^{(b)}(D_{b,j})\circ \text{circ}(X_i^{(b)})w_{1b}\|_2 \leq \beta \forall b \in [B]$. 

We can also define sets $C_{b,j} := \{ Z = u^T \in \mathbb{R}^{s \times dc} : K_{b,j} u \geq 0, \|Z\|_* \leq 1 \}$. Then, we have

$$p_{BFN}^* = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \tag{116}$$
s.t. $\max_{Z \in C_{b,j}} \text{trace} \left( \sum_{i=1}^n V_i^{(b)}(D_{b,j})\circ \text{circ}(X_i^{(b)})Z \right) \leq \beta \forall b \in [B]$. 

We form the Lagrangian as

$$p_{BFN}^* = \max_{V_i} \min_{\lambda \geq 0} \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) + \sum_{b=1}^B \sum_{j=1}^{P_b} \lambda_{b,j}(\beta - \text{trace}(Z_{b,j}^T \sum_{i=1}^n D_{b,j}^{(i)}\circ \text{circ}(X_i^{(b)}))^T V_i^{(b)})). \tag{117}$$
Table 2. Parameter count for convex heads used for the experiments in Table 1.

| CONVEX HEAD   | ACT. | PARAMS |
|---------------|------|--------|
| SELF-ATTENTION| 50.5M|        |
| MLP-MIXER     | 98.9M|        |
| B-FNO         | 392K |        |
| FNO           | 1.96M|        |
| MLP           | 1.96M|        |
| LINEAR        | 10K  |        |
| SELF-ATTENTION| 253M |        |
| MLP-MIXER     | 196M |        |
| B-FNO         | 39.2M|        |
| FNO           | 196M |        |
| MLP           | 196M |        |

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( \mathbf{V}_i \)

\[
p_{BFN}^* = \min_{\lambda_j \geq 0} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{P_1} \lambda_{1,j} \mathbf{D}_{1,j}, \text{circ}(\mathbf{X}^{(1)}_i) \mathbf{Z}_{1,j} \right) \cdots \sum_{j=1}^{P_B} \lambda_{B,j} \mathbf{D}_{B,j}, \text{circ}(\mathbf{X}^{(B)}_i) \mathbf{Z}_{B,j} \right], \mathbf{Y}_i) + \beta \sum_{b=1}^{B} \sum_{j=1}^{P_b} \lambda_{b,j}. \tag{118}
\]

Lastly, we rescale \( \tilde{Z}_{b,j} = \lambda_{b,j} \mathbf{Z}_{b,j} \) to obtain

\[
p_{BFN}^* = \min_{\mathbf{Z}_{b,j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{P_1} \mathbf{D}_{1,j}, \text{circ}(\mathbf{X}^{(1)}_i) \mathbf{Z}_{1,j} \right) \cdots \sum_{j=1}^{P_B} \mathbf{D}_{B,j}, \text{circ}(\mathbf{X}^{(B)}_i) \mathbf{Z}_{B,j} \right], \mathbf{Y}_i) + \beta \sum_{b=1}^{B} \sum_{j=1}^{P_b} \| \mathbf{Z}_{j,b} \|_*, K_{b,j}. \tag{119}
\]

as desired.

B. Experimental Details

All heads were trained on two NVIDIA 1080 Ti GPUs using the Pytorch deep learning library (Paszke et al., 2019). For our backbone, we used pre-trained weights from the Pytorch Image Models library (Wightman, 2019). For all experiments, we trained each head for 70 epochs, and used a regularization parameter of \( \beta = 2 \times 10^{-2} \), the AdamW optimizer (Loshchilov & Hutter, 2017), and a cosine learning rate schedule with a warmup of three epochs with warmup learning rate of \( 2 \times 10^{-7} \), an initial learning rate chosen based on training accuracy of either \( 5 \times 10^{-3} \) or \( 10^{-4} \), and a final learning rate of \( 2 \times 10^{-2} \) times the initial learning rate. Data augmentation was performed using AutoAugment (Cubuk et al., 2018), along with color jittering, label smoothing, and training data interpolation. All heads aside from the self-attention head were trained using a batch size of 100, whereas the self-attention head was trained with a batch size of 20. For all ReLU heads, we choose a number of neurons such that the number of parameters across FNO, MLP-Mixer, self-attention, and MLPs are roughly equal. We provide information about the number of parameters in each head in Table 2.

As an ablation, we also studied the effect of changing the backbone architecture on the results of this experiment. In particular, as a backbone, we also tried using a ViT-base model (Dosovitskiy et al., 2020) with \( 16 \times 16 \) patches pre-trained on ImageNet-1k images of size \( 224 \times 224 \) \((s = 196, d = 768)\). Then, we followed the same average pooling approach as for the gMLP backbone experiments, and kept all other network parameters the same. The results of this experiment are summarized in Table 4.

We see from this table that some of the general results from Table 1 still hold, though in this case MLP-Mixer architectures and self-attention architectures are roughly equivalent in performance. We suspect that one major reason for the larger gap between the two methods in Table 1 could be due to the backbone architecture, since the gMLP architecture is MLP-based, as opposed to ViT which is self-attention based. Thus, we speculate that adding an additional MLP-Mixer head to gMLP may be more concordant with the features extracted from the gMLP backbone, whereas the inverse is true for the ViT backbone.
We further compare the non-convex versions of some of the architectures with their convex counterparts, to demonstrate that there is no additional benefit of using a non-convex architecture. For simplicity we use ReLU rather than gated ReLU activations for the non-convex variants, to demonstrate that gated ReLU has little effect on performance. Since MLP and convolutional-based architectures have previously been shown to have an empirical equivalence in performance (Pilanci & Ergen, 2020; Ergen & Pilanci, 2021a), we focus on Self-Attention and MLP-Mixer heads. We use identical hyper-parameters for the non-convex variants of these architectures as the chosen hyper-parameters for the convex networks.

In Table 3, we observe that the performance of the non-convex heads is roughly equivalent to those of the convex heads.

| HEAD          | ACT.  | Top-1          | Top-5          |
|---------------|-------|----------------|----------------|
| SELF-ATTENTION| LINEAR| 73.81/73.72    | 92.87/92.74    |
| MLP-MIXER     |       | 78.11/77.89    | 94.79/94.99    |
| SELF-ATTENTION| (GATED)RELU | 74.74/75.32    | 93.45/93.77    |
| MLP-MIXER     |       | 80.22/77.0     | 95.79/94.53    |

We note that the main benefits of convex head are the reasons listed in the previous comment. The objective of our experiments is not to definitively state improved accuracy of convex heads over non-convex heads; this is only a single experiment, and more evaluation is required to make strong claims.

In order to avoid the heavy computational cost of nuclear norm minimization for all considered convex models (besides “linear”, which is just a logistic regression with standard weight decay), we rely on the Burer-Monteiro factorization (Burber & Monteiro, 2005). In particular, if one has the problem

$$\min_{Z \in \mathbb{R}^{a \times c}} \sum_{i=1}^{n} \mathcal{L}(f_i(Z), Y_i) + \beta \|Z\|_*,$$

we know that this is equivalent to

$$\min_{U \in \mathbb{R}^{a \times b}, V \in \mathbb{R}^{c \times b}} \sum_{i=1}^{n} \mathcal{L}(f_i(UV^\top), Y_i) + \frac{\beta}{2} \left(\|U\|_{F}^2 + \|V\|_{F}^2\right),$$

granted that the solution $Z^*$ to (120) has a rank less than the new latent dimension $b$ (Recht et al., 2010). However, while (120) is convex, (121) is not. One can also show that if the solution $Z^*$ to (120) has a rank less than the new latent dimension $b$, the problem (121) has no spurious local minima (Burber & Monteiro, 2005). We can also show that any stationary point $Z = UV^\top$ of (121), achieved e.g., with gradient descent, is a global optimum for (120) if it satisfies the following qualification condition

$$\|\sum_{i=1}^{n} \nabla_{Z} \mathcal{L}(f_i(Z), Y_i)\|_2 \leq \beta,$$

see (Mardani et al., 2013; 2015) for the proof and more details. We thus employ the Burer-Monteiro factorization for all problems, except linear. For the transformer architectures, we choose the latent dimension such that the total model size is unchanged, whereas for the MLP architecture, we choose the latent dimension to have parameters on the same order as the other transformer architectures (e.g. $b = \frac{sc}{2}$).

C. Additional Theoretical Results

C.1. Visualizing the Constrained Nuclear Norm

Here, we seek to provide some additional intuition around the constrained nuclear norm $\|Z\|_{*, K}$ for a simple case, to contrast the regularization for convex ReLU against convex linear and gated ReLU networks. We provide a visualization for
Table 4. CIFAR-100 classification accuracy for training a single convex head. Embeddings are generated from gMLP-S pre-trained on ImageNet. Note that the backbone is not fine-tuned.

| Convex Head | Act. | Top-1 | Top-5 |
|-------------|------|-------|-------|
| SELF-ATTENTION | | 67.24 | 88.63 |
| MLP-Mixer | | 66.55 | 88.70 |
| B-FNO | LINEAR | 36.04 | 64.28 |
| FNO | | 63.88 | 87.20 |
| MLP | | 56.68 | 81.79 |
| LINEAR | | 57.00 | 81.96 |
| SELF-ATTENTION | | 68.16 | 88.74 |
| MLP-Mixer | | 67.84 | 89.24 |
| B-FNO | GATED ReLU | 66.66 | 87.93 |
| FNO | | 67.87 | 88.97 |
| MLP | | 64.14 | 86.57 |

Figure 2. Visualization of the constrained nuclear norm $\|Z\|_{*,K}$ defined in Eq. (4). Here, we visualize $Z = \begin{bmatrix} z_1 & z_2 \\ z_3 & z_4 \end{bmatrix} \in \mathbb{R}^{2 \times 2}$ in terms of three of its coordinates, $z_1$, $z_2$, and $z_4$, with the final coordinate $z_3$ fixed as 0. (a) Constrained nuclear norm space arises in the convex formulation of ReLU networks (see (4)). Here, the dotted green region illustrates the set $C' = \{Z = uv^\top : u \geq 0, \|Z\|_* \leq 1, v \in \mathbb{R}^c\}$, and the opaque orange region shows its convex hull $\text{conv}(C')$. (b) In the case of linear activation, the constrained nuclear norm reduces to the standard nuclear norm $\|Z\|_*$. Notice that introducing ReLU nonlinearity breaks the symmetry present in (b) and yields a complicated non-convex space, i.e., green dots in (a). However, our convex analytic approach relaxes this set as the convex hull, i.e., orange solid space in (a), by keeping the extreme points, which are the points that play a crucial role in the optimal solution, intact. Therefore, we are able to convert standard non-convex ReLU network training problems into polynomial-time trainable convex optimization problems.

$\|Z\|_{*,K}$ compared to $\|Z\|_*$ in the case that $K = I$. This would occur in a ReLU network when $X = I$ and we encounter a particular hyperplane arrangement $D_j = I$. One can also note that in ReLU MLPs where $n \leq d$ and the data $X$ is whitened, the convex optimization objective will reduce to a linear model with this norm as its regularization (Sahiner et al., 2020).

In particular, in the case that $n = d = c = 2$, we can visualize the coordinates of $Z \in \mathbb{R}^{2 \times 2}$ which satisfy $\|Z\|_{*,K} \leq 1$, contrasted with the coordinates of $Z$ which satisfy $\|Z\|_* \leq 1$. We do so in Figure 2, illustrating the complicated, yet still convex, regularization in the case of a ReLU neural network.
C.2. Gated ReLU activation extensions

C.2.1. Self-Attention

**Theorem C.1.** For the Gated ReLU activation multi-head self-attention training problem (12), we define

\[
X := \begin{bmatrix} X_1 \otimes X_1 \\ \vdots \\ X_n \otimes X_n \end{bmatrix}
\]

\[
\{D_j\}_{j=1}^m := \{\text{diag}(I \{X_h \geq 0\})\}_{j=1}^m,
\]

for fixed gates \(\{h_j \in \mathbb{R}^d\}_{j=1}^m\). Then, the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[
p^*_S = \min_{Z_j \in \mathbb{R}^{d \times c}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \sum_{k=1}^d \sum_{\ell=1}^d G_{i,j}^{(k,\ell)} X_i Z_j^{(k,\ell)} + Y_i \right) + \beta \sum_{j=1}^m \|Z_j\|_*,
\]

where

\[
G_{i,j} := (X_i \otimes I_s)^\top D_j (X_i \otimes I_s),
\]

for \(G_{i,j}^{(k,\ell)} \in \mathbb{R}^{s \times s}\) and \(Z_j^{(k,\ell)} \in \mathbb{R}^{d \times c}\).

We note here that instead of the constrained nuclear norm penalty, we have a standard nuclear norm penalty on \(Z\).

**Proof.** We apply Lemmas A.1 and A.2 to (12) with the gated ReLU activation function to obtain

\[
p^*_S = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)
\]

s.t. \[
\max_{\|w_j\|_F \leq 1} \left\| \sum_{i=1}^N (V_i^\top \otimes V_i^\top) D_j^{(i)} (X_i \otimes X_i) w_i \right\|_2 \leq \beta \forall j \in [m].
\]

(124)

We again apply \(\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)\) (Magnus & Neudecker, 2019) to obtain

\[
p^*_S = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)
\]

s.t. \[
\max_{\|w_j\|_2 \leq 1} \left\| \sum_{i=1}^N (X_i^\top \otimes V_i^\top) D_j^{(i)} (X_i \otimes X_i) w_i \right\|_2 \leq \beta \forall j \in [m].
\]

(125)

Now, using the concept of dual norm, this is equal to

\[
p^*_S = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)
\]

s.t. \[
\max_{\|g\|_2 \leq 1} \sum_{i=1}^N (X_i^\top \otimes V_i^\top) D_j^{(i)} (X_i \otimes X_i) w_{1j} \leq \beta
\]

(126)

Then, we have

\[
p^*_S = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)
\]

s.t. \[
\max_{j \in [m]} \text{trace} \left( \sum_{i=1}^n (X_i^\top \otimes V_i^\top) D_j^{(i)} (X_i \otimes X_i) Z \right) \leq \beta
\]

(127)
Unraveling Attention via Convex Duality

Now, we simply need to form the Lagrangian and solve. The Lagrangian is given by

\[
p_{SA}^* = \max_{\lambda \geq 0} \min_{\|z_j\|_* \leq \lambda} \sum_{i=1}^n \mathcal{L}(v_i, y_i) + \sum_{j=1}^m \lambda_j \left( \beta - \sum_{i=1}^n \text{vec} \left( D_j^{(i)} (x_i \otimes z_j) \right)^\top \text{vec} (x_i \otimes v_i) \right)
\]  

(128)

We now can switch the order of max and min via Sion’s minimax theorem and maximize over \(v_i\). Defining \(K_{c,s}\) as the \((c,s)\) commutation matrix (Magnus & Neudecker, 2019):

\[
\text{vec}(x_i \otimes v_i) = ((I_d \otimes K_{c,s})(\text{vec}(x_i) \otimes I_c) \otimes I_s) \text{vec}(v_i)
\]

Maximizing over \(v_i\), we have

\[
p_{SA}^* = \min_{\lambda \geq 0} \min_{\|z_j\|_* \leq \lambda} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \left( (\text{vec}(x_i)^\top \otimes I_c)(I_d \otimes K_{s,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)} (x_i \otimes z_j) \right), \text{vec}(y_i) \right) + \beta \sum_{j=1}^m \lambda_j.
\]

(129)

Again rescaling \(\hat{z}_j = \lambda_j z_j\), we have

\[
p_{SA}^* = \min_{z_j \in \mathbb{R}^{d_x \times d_c}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \left( (\text{vec}(x_i)^\top \otimes I_c)(I_d \otimes K_{s,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)} (x_i \otimes z_j) \right), \text{vec}(y_i) \right) + \beta \sum_{j=1}^m \|z_j\|_*. 
\]

(130)

It appears as though this is a very complicated function, but it actually simplifies greatly. In particular, one can write this as

\[
p_{SA}^* = \min_{z_j \in \mathbb{R}^{d_x \times d_c}} \sum_{i=1}^n \mathcal{L} \left( \hat{y}_i, y_i \right) + \beta \|z_j\|
\]

\[
\hat{y}_i[0, p] := \sum_{j=1}^m \sum_{k=1}^d \sum_{l=1}^s x_i[t, l] x_i[t, k] d_j^{(t,m)} x_i[0, :]^\top z_j^{(k,l)}.
\]

(131)

Making any final simplifications, one obtains the desired result.

\[
p_{SA}^* = \min_{z_j \in \mathbb{R}^{d_x \times d_c}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \sum_{k=1}^d \sum_{l=1}^s g_j^{(k,l)} x_i z_j^{(k,l)} y_i \right) + \beta \sum_{j=1}^m \|z_j\|_*. 
\]

(132)

C.2.2. MLP-MIXER

Theorem C.2. For the Gated ReLU activation MLP-Mixer training problem (17), we define

\[
X := \begin{bmatrix} x_i^\top \otimes I_s \\ \cdots \\ x_i^\top \otimes I_s \end{bmatrix}
\]

\[
\{D_j\}_{j=1}^m := \{\text{diag}(\{1 \{x_h \geq 0\}\})\},
\]

for fixed gates \(\{h_j \in \mathbb{R}^{s^2}\}_{j=1}^m\). Then, the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[
p_{MM}^* = \min_{z_j \in \mathbb{R}^{s \times d_c}} \sum_{i=1}^n \mathcal{L} \left( \left[ f_1(x_i) \cdots f_c(x_i) \right], y_i \right) + \beta \sum_{j=1}^m \|z_j\|_*
\]

(133)

where

\[
f_p(x_i) := \sum_{j=1}^m \left[ D_j^{(i,1)} z_j^{(p,1)} \cdots D_j^{(i,d)} z_j^{(p,d)} \right] \text{vec}(x_i)
\]

for \(D_j^{(i,k)} \in \mathbb{R}^{s \times s}\) and \(z_j^{(p,k)} \in \mathbb{R}^{s \times s}\).
Proof. We apply Lemmas A.1 and A.2 to (17) with the Gated ReLU activation function to obtain

\[
p^*_MM = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^n V_i^T \sigma_j (W_i X_i) \right\|_F \leq \beta \tag{134}
\]

This is equivalent to (Magnus & Neudecker, 2019)

\[
p^*_MM = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|W_i\|_F \leq 1} \left\| \sum_{i=1}^n (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_s) \text{vec}(W_i) \right\|_F \leq \beta \tag{135}
\]

Now, using the concept of dual norm, this is equal to

\[
p^*_MM = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|g\|_2 \leq 1} \max_{\|W_i\|_F \leq 1} \sum_{j \in [m]} \mathbf{g}^T \left( \sum_{i=1}^n (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_s) \right) w_i \leq \beta \tag{136}
\]

Then, we have

\[
p^*_MM = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) \\
\text{s.t. } \max_{\|Z\|_* \leq 1} \sum_{j \in [m]} \operatorname{trace} \left( \sum_{i=1}^n (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_s) Z \right) \leq \beta \tag{137}
\]

Now, we simply need to form the Lagrangian and solve. The Lagrangian is given by

\[
p^*_MM = \max_{V_i} \min_{\lambda \geq 0} \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) + \sum_{j=1}^m \lambda_j \left( \beta - \sum_{i=1}^n \operatorname{trace} \left( (I_d \otimes V_i^T) D_j^{(i)} (X_i^T \otimes I_s) Z \right) \right) \tag{138}
\]

We now can switch the order of max and min via Sion’s minimax theorem and maximize over \(V_i\):

\[
p^*_MM = \min_{\lambda \geq 0} \min_{\|Z\|_* \leq 1} \max_{V_i} \sum_{i=1}^n \mathcal{L}^* (V_i, Y_i) + \sum_{j=1}^m \lambda_j \left( \beta - \sum_{i=1}^n \operatorname{vec} \left( D_j^{(i)} (X_i^T \otimes I_s) Z \right)^T \operatorname{vec} (I_d \otimes V_i) \right) \tag{139}
\]

Now, defining \(K_{c,d}\) as the \((c, d)\) commutation matrix:

\[
\operatorname{vec}(I_d \otimes V_i) = (I_d \otimes K_{c,d})(\operatorname{vec}(I_d) \otimes I_c) \otimes I_s) \operatorname{vec}(V_i)
\]

Solving over \(V_i\) yields

\[
p^*_MM = \min_{\lambda \geq 0} \min_{\|Z\|_* \leq 1} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \left( \operatorname{vec}(I_d)^T \otimes I_c) (I_d \otimes K_{d,c}) \otimes I_s) \operatorname{vec} \left( D_j^{(i)} (X_i^T \otimes I_s) \lambda_j Z_j \right), \operatorname{vec}(Y_i) \right) + \beta \sum_{j=1}^m \lambda_j \tag{140}
\]
Re-scaling $Z_j = \lambda_j Z_j$ gives us

$$p^*_{MM} = \min_{Z_j} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \left( \text{vec}(I_d^T \otimes I_c)(I_d \otimes K_{d,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)}(X_i^T \otimes I_s)Z_j \right), \text{vec}(Y_i) \right) + \beta \sum_{j=1}^m \|Z_j\|_*.$$  

(141)

One can actually greatly simplify this result, and can re-write this as

$$p^*_{MM} = \min_{Z_j \in \mathbb{R}^{s \times dc}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m \left( \text{vec}(I_d^T \otimes I_c)(I_d \otimes K_{d,c}) \otimes I_s \right) \text{vec} \left( D_j^{(i)}(X_i^T \otimes I_s)Z_j \right), \text{vec}(Y_i) \right) + \beta \sum_{j=1}^m \|Z_j\|_*.$$  

(142)

$$f_p(X_i) := \sum_{j=1}^m \left[ f_j^{(1)} \circ \cdots \circ f_j^{(c)}(X_i) \right] \text{vec}(X_i).$$  

(143)

as desired.

C.2.3. FNO

**Theorem C.3.** For the Gated ReLU activation FNO training problem (24), we define

$$X := \begin{bmatrix} \text{circ}(X_1) \\ \vdots \\ \text{circ}(X_n) \end{bmatrix}$$

$$\{D_j\}_{j=1}^m := \{\text{diag}(I \{Xh_j \geq 0\})\},$$

for fixed gates $\{h_j \in \mathbb{R}^{sd} \}_{j=1}^m$. Then, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p^*_{FN} = \min_{Z_j \in \mathbb{R}^{s \times dc}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m D_j^{(i)} \text{circ}(X_i)Z_j, Y_i \right) + \beta \sum_{j=1}^m \|Z_j\|_*.$$  

(144)

**Proof.** We now apply Lemmas A.1 and A.2 with the Gated ReLU activation function to obtain

$$p^*_{FN} = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t.} \quad \max_{\|w\|_2 \leq 1} \sum_{i=1}^n V_i^T D_j^{(i)} \text{circ}(X_i)w \leq \beta.$$  

(145)

Using the concept of dual norm, this is equivalent to

$$p^*_{FN} = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t.} \quad \max_{\|g\|_2 \leq 1} \sum_{i=1}^n V_i^T D_j^{(i)} \text{circ}(X_i)w \leq \beta.$$  

(146)

Then, we have

$$p^*_{FN} = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i)$$

$$\text{s.t.} \quad \max_{\|Z\|_* \leq 1} \text{trace} \left( \sum_{i=1}^n V_i^T D_j^{(i)} \text{circ}(X_i)Z \right) \leq \beta.$$  

(147)
Unraveling Attention via Convex Duality

We form the Lagrangian as

\[ p^*_N = \max_{V_i} \min_{\lambda \geq 0} \min_{\|Z_j\|_* \leq 1} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^m \lambda_j (\beta - \text{trace}(Z_j^\top \sum_{i=1}^n D_j^{(i)} \circ(X_i^\top V_i))). \]  

(148)

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( V_i \)

\[ p^*_N = \min_{\lambda \geq 0} \min_{\|Z_j\|_* \leq 1} \sum_{i=1}^n \mathcal{L}(\sum_{j=1}^m D_j^{(i)} \circ(X_i, Y_i)) + \beta \sum_{j=1}^m \lambda_j. \]  

(149)

Lastly, we rescale \( \tilde{Z}_j = \lambda_j Z_j \) to obtain

\[ p^*_N = \min_{Z_j \in \mathbb{R}^{sd \times c}} \sum_{i=1}^n \mathcal{L} \left( \sum_{j=1}^m D_j^{(i)} \circ(X_i, Y_i) \right) + \beta \sum_{j=1}^m \|Z_j\|_*. \]  

(150)

as desired.

C.2.4. B-FNO

**Theorem C.4.** For the Gated ReLU activation B-FNO training problem (29), we define

\[ X_b := \begin{bmatrix} \circ(X_1^{(b)}) \\ \vdots \\ \circ(X_n^{(b)}) \end{bmatrix} \]

\( \{D_{b,j}\}_{j=1}^m := \{\text{diag}(\mathds{1}\{X_b h_{b,j} \geq 0\})\}, \)

for fixed gates \( \{h_{b,j} \in \mathbb{R}^{sd/B}\}_{j=1}^m \). Then, the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[ p^*_B = \min_{Z_{b,j}} \sum_{i=1}^n \mathcal{L} \left( \left[ f^{(1)}(X_i) \ldots f^{(B)}(X_i) \right], Y_i \right) + \beta \sum_{b=1}^B \sum_{j=1}^m \|Z_{j,b}\|_*, \]  

(151)

where

\[ f^{(b)} := \sum_{j=1}^m D_{b,j} \circ(X_i^{(b)}) Z_{j,b} \]

**Proof.** We now apply Lemmas A.1 and A.2 with the Gated ReLU activation function to obtain

\[ p^*_B = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \]

\[ \text{s.t. } \max_{\|g\|_2 \leq 1} \| \sum_{i=1}^n V_i^{(b)} \circ(X_i^{(b)}) w_{1b} \|_2 \leq \beta \]  

(152)

Using the concept of dual norm, this is equivalent to

\[ p^*_B = \max_{V_i} - \sum_{i=1}^n \mathcal{L}^*(V_i, Y_i) \]

\[ \text{s.t. } \max_{\|g\|_2 \leq 1} \| \sum_{i=1}^n V_i \circ(X_i^{(b)} D_{b,j}^{(i)}) w_{1b} \|_2 \leq \beta \forall b \in [B]. \]  

(153)
We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over
the non-convex training objective is equivalent to a convex optimization problem, given by

\[
\min_{\mathbf{L}, \mathbf{w}} f_{\mathbf{P}}(\mathbf{X}_i) = \sigma(\mathbf{P}\mathbf{X}_i)\mathbf{W}_1\mathbf{W}_2
\]

where \( \mathbf{P} \in \mathbb{R}^{s \times s} \) is a local pooling function. In this way, the PoolFormer architecture still mixes across different tokens, but in a non-learnable, deterministic fashion.

In (Lee-Thorp et al., 2021), the authors propose FNet, another alternative which resembles PoolFormer architecture. In particular, a 2D FFT is applied to the input \( \mathbf{X}_i \) before being passed through an MLP

\[
f_{\text{FNET}}(\mathbf{X}_i) = \sigma(\mathbf{F}_d\mathbf{X}_i\mathbf{F}_d^\top\mathbf{W}_1)\mathbf{W}_2
\]

One can use similar convex duality results as in the main body of this paper to generate convex dual forms for this architecture for linear, ReLU, and gated ReLU activation PoolFormers and FNet. To keep these results general, we will be analyzing networks of the form

\[
p^*_{\text{PF}} := \min_{\mathbf{w}_{1j}, \mathbf{w}_{2j}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} \sigma(b(h(\mathbf{X}_i)\mathbf{W}_{1j})\mathbf{W}_{2j}^\top), \mathbf{Y}_i \right) + \frac{\beta}{2} \sum_{j=1}^{m} \|\mathbf{w}_{1j}\|^2 + \|\mathbf{w}_{2j}\|^2
\]

for any generic function \( h : \mathbb{R}^{s \times d} \to \mathbb{R}^{s \times d} \), which encapsulates both methods and more.

**Theorem C.5.** For the linear activation network training problem (160), for \( m \geq m^* \) where \( m^* \leq \min\{d, c\} \), the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[
p^*_{\text{PF}} = \min_{\mathbf{Z} \in \mathbb{R}^{d \times s}} \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{X}_i)\mathbf{Z}, \mathbf{Y}_i) + \beta\|\mathbf{Z}\|_+.
\]
Proof. We now apply Lemmas A.1 and A.2 to obtain

\[ p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

subject to

\[ \max_{\|w_1\|_2 \leq 1} \| \sum_{i=1}^{n} V_i^T h(X_i)w_1 \|_2 \leq \beta \]  

(162)

This is equivalent to

\[ p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

subject to

\[ \| \sum_{i=1}^{n} V_i^T h(X_i) \|_2 \leq \beta \]

(163)

We form the Lagrangian as

\[ p_{PF}^* = \max_{V_i} \min_{\lambda \geq 0} \min_{\|Z\|_* \leq 1} \sum_{i=1}^{n} \mathcal{L}(h(X_i)Z, Y_i) + \lambda(\beta - \text{trace}(Z^T \sum_{i=1}^{n} h(X_i)^T V_i)). \]

(164)

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( V_i \)

\[ p_{PF}^* = \min_{\lambda \geq 0} \min_{\|Z\|_* \leq 1} \sum_{i=1}^{n} \mathcal{L}(h(X_i)Z, Y_i) + \beta \lambda. \]

(165)

Lastly, we rescale \( \tilde{Z} = \lambda Z \) to obtain

\[ p_{PF}^* = \min_{Z \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L}(h(X_i)Z, Y_i) + \beta \|Z\|_* \]

(166)

as desired.

\[ \square \]

**Theorem C.6.** For the ReLU activation training problem (160), we define

\[ X := \begin{bmatrix} h(X_1) \\ \vdots \\ h(X_n) \end{bmatrix} \]

\[ \{D_j\}_{j=1}^P := \{ \text{diag}(\mathbb{1}\{Xu_j \geq 0\}) : u_j \in \mathbb{R}^d \}, \]

where \( P \leq 2r \left( \frac{e(n-1)}{r} \right)^r \) and \( r := \text{rank}(X) \). Then, for \( m \geq m^* \) where \( m^* \leq n \min\{d, c\} \), the standard non-convex training objective is equivalent to a convex optimization problem, given by

\[ p_{PF}^* = \min_{Z_j \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{P} D_j^{(i)}h(X_i)Z_j, Y_i \right) + \beta \sum_{j=1}^{P} \|Z_j\|_{*,K}, \]

(167)

where

\[ K_j := (2D_j - I_{ns})X. \]

Proof. We now apply Lemmas A.1 and A.2 with the ReLU activation function to obtain

\[ p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

subject to

\[ \max_{\|w_1\|_2 \leq 1} \| \sum_{i=1}^{n} V_i^T (h(X_i)w_1) \|_2 \leq \beta \]

(168)
We introduce hyperplane arrangements $D_j$ and enumerate over all of them, yielding

$$
p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
$$

s.t.

$$
\max_{\|w_i\|_2 \leq 1 \bigg| K_j, w_i \geq 0} \|\sum_{i=1}^{n} V_i^T D_j(i) h(X_i) w_1\|_2 \leq \beta.
$$

(169)

Using the concept of dual norm, this is equivalent to

$$
p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
$$

s.t.

$$
\max_{\|g\|_2 \leq 1 \bigg| \|w_i\|_2 \leq 1 \bigg| K_j, w_i \geq 0} g^T \sum_{i=1}^{n} V_i^T D_j(i) h(X_i) w_1 \leq \beta
$$

(170)

We can also define sets $\mathcal{C}_j := \{Z = u g^T \in \mathbb{R}^{d \times c} : K_j u \geq 0, \|Z\|_* \leq 1\}$. Then, we have

$$
p_{PF}^* = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i)
$$

s.t.

$$
\max_{j \in [1]^P \bigg| z \in \mathcal{C}_j} \left(\sum_{i=1}^{n} V_i^T D_j(i) h(X_i) Z\right) \leq \beta
$$

(171)

We form the Lagrangian

$$
p_{PF}^* = \max_{\lambda \geq 0} \min_{Z_j \in \mathcal{C}_j} \min_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^{P} \lambda_j (\beta - \text{trace}(Z_j \sum_{i=1}^{n} D_j(i) h(X_i)^T V_i)).
$$

(172)

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over $V_i$

$$
p_{PF}^* = \min_{\lambda_j \geq 0} \min_{Z_j \in \mathcal{C}_j} \sum_{i=1}^{n} \mathcal{L}(\sum_{j=1}^{P} D_j(i) h(X_i) Z_j, Y_i) + \beta \sum_{j=1}^{P} \lambda_j.
$$

(173)

Lastly, we rescale $\tilde{Z}_j = \lambda_j Z_j$ to obtain

$$
p_{PF}^* = \min_{Z_j \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L} \left(\sum_{j=1}^{P} D_j(i) h(X_i) Z_j, Y_i\right) + \beta \sum_{j=1}^{P} \|Z_j\|_* K_j.
$$

(174)

as desired. \hfill \Box

**Theorem C.7.** For the Gated ReLU activation training problem (160), we define

$$X := \begin{bmatrix} h(X_1) \\ \vdots \\ h(X_n) \end{bmatrix}$$

\{D_j\}_{j=1}^{P} := \{\text{diag}(I \{X h_j \geq 0\})\},

for fixed gates $\{h_j \in \mathbb{R}^{d}\}_{j=1}^{m}$. Then, the standard non-convex training objective is equivalent to a convex optimization problem, given by

$$p_{PF}^* = \min_{Z_j \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L} \left(\sum_{j=1}^{m} D_j(i) h(X_i) Z_j, Y_i\right) + \beta \sum_{j=1}^{m} \|Z_j\|_*
$$

(175)
Proof. We now apply Lemmas A.1 and A.2 with the Gated ReLU activation function to obtain

\[ p^*_PF = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

s.t. \[ \max_{\|w_i\|_2 \leq 1} \sum_{i=1}^{n} \left\| V_i^T D_j^{(i)} h(X_i) w_1 \right\|_2 \leq \beta. \] (176)

Using the concept of dual norm, this is equivalent to

\[ p^*_PF = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

s.t. \[ \max_{\|g\|_2 \leq 1} \sum_{i=1}^{n} g_i^T V_i^T D_j^{(i)} h(X_i) w_1 \leq \beta \] (177)

Then, we have

\[ p^*_PF = \max_{V_i} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) \]

s.t. \[ \max_{\|Z\|_* \leq 1} \text{trace}\left( \sum_{i=1}^{n} V_i^T D_j^{(i)} h(X_i) Z \right) \leq \beta \] (178)

We form the Lagrangian as

\[ p^*_PF = \max_{V_i} \min_{\lambda_j \geq 0} \min_{\|Z\|_* \leq 1} - \sum_{i=1}^{n} \mathcal{L}^*(V_i, Y_i) + \sum_{j=1}^{m} \lambda_j (\beta - \text{trace}(Z_j^T \sum_{i=1}^{n} D_j^{(i)} h(X_i)^T V_i)). \] (179)

We switch the order of the maximum and minimum using Sion’s minimax theorem and maximize over \( V_i \)

\[ p^*_PF = \min_{\lambda_j \geq 0} \min_{\|Z\|_* \leq 1} \sum_{i=1}^{n} \mathcal{L}(\sum_{j=1}^{m} D_j^{(i)} h(X_i) Z_j, Y_i) + \beta \sum_{j=1}^{m} \lambda_j. \] (180)

Lastly, we rescale \( Z_j = \lambda_j Z_j \) to obtain

\[ p^*_PF = \min_{Z_j \in \mathbb{R}^{d \times c}} \sum_{i=1}^{n} \mathcal{L} \left( \sum_{j=1}^{m} D_j^{(i)} h(X_i) Z_j, Y_i \right) + \beta \sum_{j=1}^{m} \|Z_j\|_* \] (181)

as desired. \qed