# Positive Concave Deep Equilibrium Models

**Mateusz Gabor** [1] **Tomasz Piotrowski** [2] **Renato L. G. Cavalcante** [3]

## Abstract

Deep equilibrium (DEQ) models are widely recognized as a memory efficient alternative to standard neural networks, achieving state-of-the-art performance in language modeling and computer vision tasks. These models solve a fixed point equation instead of explicitly computing the output, which sets them apart from standard neural networks. However, existing DEQ models often lack formal guarantees of the existence and uniqueness of the fixed point, and the convergence of the numerical scheme used for computing the fixed point is not formally established. As a result, DEQ models are potentially unstable in practice. To address these drawbacks, we introduce a novel class of DEQ models called positive concave deep equilibrium (pcDEQ) models. Our approach, which is based on nonlinear Perron-Frobenius theory, enforces nonnegative weights and activation functions that are concave on the positive orthant. By imposing these constraints, we can easily ensure the existence and uniqueness of the fixed point without relying on additional complex assumptions commonly found in the DEQ literature, such as those based on monotone operator theory in convex analysis. Furthermore, the fixed point can be computed with the standard fixed point algorithm, and we provide theoretical guarantees of its geometric convergence, which, in particular, simplifies the training process. Experiments demonstrate the competitiveness of our pcDEQ models against other implicit models.

[1]Faculty of Electronics, Photonics, and Microsystems,Wrocław University of Science and Technology, Wrocław, Poland [2]Faculty of Physics, Astronomy and Informatics, Nicolaus Copernicus University, Toruń, Poland [3]Fraunhofer Heinrich Hertz Institute, Berlin, Germany. Correspondence to: Mateusz Gabor <mateusz.gabor@pwr.edu.pl>.

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

## 1. Introduction

Implicit models (Bai et al., 2019; 2020; Chen et al., 2018; El Ghaoui et al., 2021; Baker et al., 2023; Tsuchida & Ong, 2023; Revay et al., 2020; Wei & Kolter, 2021; Geng et al., 2021) are attracting considerable interest owing to their improved memory efficiency compared to standard neural networks. These models solve implicit equations instead of explicitly computing the output of the layers, and they can be divided into two main categories: neural ordinary differential equations (Neural ODEs) (Chen et al., 2018) and deep equilibrium models (DEQ) models (Bai et al., 2019).

Neural ODEs solve differential equations parameterized by the neural network input, with the output representing the solution to these equations. On the other hand, the implicit layers of DEQ models solve nonlinear fixed point equations that are not necessarily derived from differential equations. An interesting aspect about DEQ models is that a single implicit DEQ layer emulates a standard neural network with an infinite number of layers and tied weights. While both DEQ models and neural ODEs require constant training memory, DEQ models often outperform neural ODEs, achieving state-of-the-art results in language modeling tasks (Bai et al., 2019) and computer vision tasks (Bai et al., 2020; Xie et al., 2022). However, a potential limitation of standard DEQ models is that they are based on iterative methods that require careful initialization, tuning of hyperparameters, and special regularization (e.g., recurrent dropout) to ensure convergence to the fixed point, which is hard to guarantee with existing approaches. In general, standard DEQ models operate heuristically, and in many cases they even lack formal guarantees regarding the existence and uniqueness of the fixed point problem being solved.

To overcome the limitations of existing DEQ models, we introduce a new variant called positive concave deep equilibrium (pcDEQ) models. In particular, these pcDEQ models address issues related to the existence and uniqueness of the fixed point. Furthermore, the fixed points of the proposed pcDEQ models can be easily computed with the standard fixed point iteration, and we provide formal guarantees of geometric convergence. The theoretical foundation of pcDEQ models is rooted in nonlinear Perron-Frobenius (NPF) theory (Lemmens & Nussbaum, 2012; Lins, 2023), which is commonly used in the analysis of nonnegative

monotonic (order-preserving) and scalable functions. To ensure these properties in pcDEQ models, we enforce nonnegative weights and activation functions that are concave in the nonnegative orthant. An additional advantage of our proposed model is that the standard Jacobian-based backpropagation algorithm (Bai et al., 2019) can be used for training without requiring any changes.

We summarize our contributions as follows:

1. We propose a new class of DEQ models, called pcDEQ models, which are based on nonlinear Perron-Frobenius theory and are equipped with guarantees of the existence and uniqueness of the fixed point. Furthermore, we prove that the standard fixed point iteration for the proposed pcDEQ models converges to the fixed point geometrically fast.

2. We empirically show that, for the proposed pcDEQ architectures, in practice, only a few iterations are needed to achieve numerical convergence, and the number of iterations does not increase over the course of training.

3. We demonstrate competitive improvement of the proposed approach in terms of accuracy and number of parameters over existing alternatives for image classification tasks.

## 2. Related Work

In the seminal paper (Bai et al., 2019), DEQ models have been applied to language modeling tasks, and they have been shown to outperform standard neural networks constructed with a similar number of parameters. In subsequent studies, Bai *et al.* proposed *multiscale* extensions of DEQ models (Bai et al., 2020), where a single DEQ model is used for image classification and image segmentation. Since these pioneering studies, DEQ models have been successful in many applications, including, to name a few, object detection (Wang et al., 2023), optical flow estimation (Bai et al., 2022), video semantic segmentation (Ertenli et al., 2022), medical image segmentation (Zhang et al., 2022), snapshot compressive imaging (Zhao et al., 2023), image denoising (Chen et al., 2023; Gkillas et al., 2023a), machine translation (Zheng et al., 2023), inverse problems (Gilton et al., 2021; Zou et al., 2023), music source separation (Koyama et al., 2022), federated learning (Gkillas et al., 2023b), and diffusion models (Geng et al., 2023; Pokle et al., 2022). These models are strongly based on the fixed point theory that we summarize below.

Let $\mathbb{R}^n$ be the standard Euclidean metric space and let $f : \mathbb{R}^n \to \mathbb{R}^n$. If $f$ is a Lipschitz contraction, meaning it is Lipschitz continuous with a Lipschitz constant $L < 1$, then the Banach fixed point theorem guarantees the uniqueness and existence of the fixed point $x^\star = f(x^\star)$ in $\mathbb{R}^n$. Moreover, the fixed point iteration $x_{k+1} = f(x_k)$ converges linearly to $x^\star$ for any initial point $x_1 \in \mathbb{R}^n$. However, if $f$ is not a Lipschitz contraction, questions related to the existence and uniqueness of the unique point, and also the convergence of the fixed point iteration, become more delicate, and these issues have been the subject of extensive research in the mathematical literature.

In modern convex analysis, there has been a significant focus on nonexpansive mappings in Hilbert spaces (Yamada et al., 2011), which are mappings with Lipschitz constant equal to one, and their relation to monotone operator theory (Bauschke & Combettes, 2017). Nonexpansive mappings do not necessarily have a fixed point, and, if the fixed point set is nonempty, it may not be a singleton in general. Furthermore, even if the fixed point exists, then the fixed point iteration may fail to converge, in which case one can resort to various iterative methods based on the the Krasnosel'skii-Mann iteration to ensure it. Particular instances of this iteration include well-known algorithms used in machine learning, such as the projected gradient method, the proximal forward-backward splitting method, the Douglas-Rachford splitting method, the projection onto convex sets method, and many others (Yamada et al., 2011).

Monotone operator theory in DEQ models has been explored in (Winston & Kolter, 2020), a study that introduces the monotone operator deep equilibrium (monDEQ) models. These models ensure both the existence and uniqueness of the fixed point, while also guaranteeing the convergence of the forward-backward splitting algorithm and Peaceman-Rachford splitting algorithm to the fixed point. However, approaches of this type require restrictions on weights that can only be enforced with complex numerical techniques.

To avoid the above difficulties, we use in this study nonlinear Perron-Frobenius theory (Krause, 2015; Lemmens & Nussbaum, 2012) as an alternative to traditional theory in Hilbert spaces. More specifically, we consider deep equilibrium layers that belong to the class of standard interference mappings. These mappings, introduced in the next section, have been widely used in wireless networks (Yates, 1995; Schubert & Boche, 2011; Stanczak et al., 2009; You & Yuan, 2020; Miretti et al., 2023a;b; Shindoh, 2020; 2019; Cavalcante et al., 2016; 2019). They are not necessarily nonexpansive in Hilbert spaces, but they are contractive (though not necessarily Lipschitz contractions) in some metric spaces defined on cones, such as the cone of positive vectors. Before introducing SI mappings and their applications in DEQ models, we first need to establish the notation and formally introduce the concept of DEQ layers.

## 3. Preliminaries

The nonnegative cone and its interior (i.e., the positive cone) are denoted as, respectively,

$$\mathbb{R}^n_+ := \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid (\forall k \in \{1, \ldots, n\})\ x_k \geq 0\}.$$

and

$$\operatorname{int}(\mathbb{R}^n_+) := \{(x_1, \ldots, x_n) \in \mathbb{R}^n_+ \mid (\forall k \in \{1, \ldots, n\})\ x_k > 0\}.$$

Let $x, y \in \mathbb{R}^n_+$. The partial ordering induced by the nonnegative cone is denoted as $x \leq y \Leftrightarrow y - x \in \mathbb{R}^n_+$. In a similar way, for $x \neq y$, $x < y \Leftrightarrow y - x \in \mathbb{R}^n_+$, and $x \ll y \Leftrightarrow y - x \in \operatorname{int}(\mathbb{R}^n_+)$. The fixed point set of a function $f : X \to Y$ with $Y$ and $X$ being subsets of a given set $S$ is denoted as

$$\operatorname{Fix}(f) = \{x^\star \in X \mid f(x^\star) = x^\star\}.$$

### 3.1. Deep Equilibrium Layers

We now have all the necessary notation to introduce generic DEQ models.

**Definition 3.1.** A DEQ layer maps an input $x \in X \subset \mathbb{R}^n$ to an output $z^\star \in \operatorname{Fix}(g_x) \subset Y \subset X$, where $g_x : X \to Y$ is an explicit function given by

$$g_x : X \to Y : z \mapsto \sigma(Wz + x); \tag{1}$$

$W\colon X \to Y$ is a linear operator (weight matrix); and $\sigma\colon X \to Y$ is a (vector-valued nonlinear) activation function, composed elementwise from a given scalar activation function.

In the above definition, closed-form expressions for the implicit function $x \mapsto z^\star \in \operatorname{Fix}(g_x)$ are not required, and the numerical scheme used to compute the output $z^\star \in \operatorname{Fix}(g_x)$ from a given input $x$ is not specified. As a result, to ensure that the implicit function $x \mapsto z^\star \in \operatorname{Fix}(g_x)$ is well-defined and independent of the numerical scheme used to compute the output, we require $\operatorname{Fix}(g_x)$ to be a singleton for every $x \subset X$. This fixed point formulation allows for direct implicit differentiation, which is crucial for training DEQ models (Bai et al., 2019). In the text that follows, for convenience, we always refer to a DEQ layer using its explicit function $g_x$ because, with the above restriction of $\operatorname{Fix}(g_x)$ being a singleton, the implicit function $x \mapsto z^\star \in \operatorname{Fix}(g_x)$ is well-defined.

One of the simplest numerical schemes for computing fixed points, which is the numerical scheme we consider in this study, is the standard fixed point iteration, which, using the notation in Definition 3.1, we can write as

$$(\forall k \in \mathbb{N})\ z_{k+1} = g_x(z_k) \text{ with } z_1 \in X. \tag{2}$$

### 3.2. Standard Interference and Positive Concave Mappings

The DEQ layers that we propose in this study are a proper subclass of standard interference mappings, defined as follows.

**Definition 3.2.** A mapping $g\colon \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ is said to be a standard interference if it is

1. monotonic

$$(\forall x \in \mathbb{R}^n_+)(\forall \tilde{x} \in \mathbb{R}^n_+)\ x \leq \tilde{x} \implies g(x) \leq g(\tilde{x}), \text{ and} \tag{3}$$

2. scalable

$$(\forall x \in \mathbb{R}^n_+)(\forall \lambda > 1) \quad g(\lambda x) \ll \lambda g(x). \tag{4}$$

*Remark* 3.3. Monotonic mappings in the sense of Definition 3.2 are also known as order-preserving mappings in the mathematical literature, and they should not be confused with monotone operators used in convex analysis (Ryu & Boyd, 2016; Bauschke et al., 2017), which is a different concept in general.

SI mappings have at most one fixed point (Yates, 1995), and its existence can be established with Proposition A.4 in the appendix, which uses the concepts of asymptotic mappings (Definition A.2) and nonlinear spectral radius (Definition A.3). A proper subclass of SI mappings is the class of the positive concave mappings (see Proposition A.6), defined below.

**Definition 3.4.** Let $g : \mathbb{R}^n_+ \to \mathbb{R}^n_+$ be concave with respect to the cone order; i.e.,

$$(\forall x \in \mathbb{R}^n_+)(\forall y \in \mathbb{R}^n_+)(\forall t \in (0,1)) \quad g(tx + (1-t)y) \geq tg(x) + (1-t)g(y). \tag{5}$$

Then $g$ is called a nonnegative concave (NC) mapping. Furthermore, if the codomain of $g$ is in the set of positive vectors (i.e., $g : \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$), then $g$ is called a positive concave (PC) mapping.

We emphasize that NC mappings are not SI mappings in general (e.g., $x \mapsto x$ for $x \in \mathbb{R}^n_+$ is an NC, but not an SI mapping), only PC mappings are guaranteed to be SI mappings without any further assumptions, see (Cavalcante et al., 2016; 2019).

An important property of PC mappings, which we exploit in this study, is that, if a fixed point exists, the fixed point iteration is guaranteed to converge geometrically fast to the fixed point (Proposition A.7).

## 4. PC Deep Equilibrium Layers

We now proceed to construct the proposed pcDEQ layers $g_x$ based on the theory described in the previous section. In particular, recall that, by restricting DEQ layers to the class of PC mappings, we obtain simple conditions to guarantee the existence and uniqueness of the fixed point $z^\star$, and we

also have the simple iterative scheme in (2) to compute the fixed point, which is an algorithm that converges geometrically fast for PC mappings. We start by restricting the activation functions to be nonnegative concave in the nonnegative cone, and, for later reference, the next remark gives examples of such functions.

*Remark* 4.1. We divide the activation functions allowed by the proposed framework into two classes: nonnegative concave functions and positive concave activation functions. Common examples in the neural network literature are shown in the two lists below.

1. continuous NC activation functions ($\mathbb{R}_+ \to \mathbb{R}_+$):
   - (ReLU6) $x \mapsto \min\{x, 6\}$
   - (hyperbolic tangent) $x \mapsto tanh\ x$
   - (softsign) $x \mapsto \frac{x}{1+x}$
2. continuous PC activation functions ($\mathbb{R}_+ \to \text{int}(\mathbb{R}_+)$):
   - (sigmoid) $x \mapsto \frac{1}{1+\exp(-x)}$

We recall that the activation functions from the above lists are applied elementwise to vector arguments, see Definition 3.1. The following lemma provides two simple sufficient conditions to construct positive concave DEQ layers.

**Lemma 4.2.** *Consider a DEQ layer $g_x \colon \mathbb{R}^n_+ \to \text{int}(\mathbb{R}^n_+)$ of the form in (1) in Definition 3.1 for an input $x$. Then:*

- *(Assumption 1) $z \mapsto g_x(z) := \sigma(Wz+x)$ is a PC mapping $g_x \colon \mathbb{R}^n_+ \to \text{int}(\mathbb{R}^n_+)$ if $z \in \mathbb{R}^n_+$, $W \in \mathbb{R}^{n\times n}_+$, $x \in \text{int}(\mathbb{R}^n_+)$, and $\sigma$ is constructed elementwise from any scalar activation function from List 1 in Remark 4.1;*
- *(Assumption 2) $z \mapsto g_x(z) := \sigma(Wz + x)$ is a PC mapping $g_x \colon \mathbb{R}^n_+ \to \text{int}(\mathbb{R}^n_+)$ if $z \in \mathbb{R}^n_+$, $W \in \mathbb{R}^{n\times n}_+$, $x \in \mathbb{R}^n_+$, and $\sigma$ is constructed elementwise from the scalar activation function from List 2 in Remark 4.1.*

*Proof.* Let $f_x \colon \mathbb{R}^n_+ \to \mathbb{R}^n_+$ be given by $z \mapsto Wz + x$ and let $\sigma \colon \mathbb{R}^n_+ \to \mathbb{R}^n_+$ be the activation function. Let $f_x$ and $\sigma$ satisfy either Assumption 1 or Assumption 2, so that $g_x = \sigma \circ f_x$. We note that both $f_x$ and $\sigma$ belong to the class of NC mappings. Let $z_1, z_2 \in \mathbb{R}^n_+$ and $t \in (0, 1)$. Then, by concavity of $f_x$:

$$f_x(tz_1 + (1-t)z_2) \geq tf_x(z_1) + (1-t)f_x(z_2).$$

Hence, monotonicity of $\sigma$ implies that

$$\sigma[f_x(tz_1 + (1-t)z_2)] \geq \sigma[tf_x(z_1) + (1-t)f_x(z_2)].$$

From the concavity of $\sigma$ we deduce

$$\sigma[tf_x(z_1)+(1-t)f_x(z_2)] \geq t\sigma(f_x(z_1))+(1-t)\sigma(f_x(z_2)).$$

By combining the above two inequalities, we conclude that

$$(\sigma\circ f_x)(tz_1+(1-t)z_2) \geq t(\sigma\circ f_x)(z_1)+(1-t)(\sigma\circ f_x)(z_2),$$

implying that $g_x = \sigma \circ f_x$ is an NC mapping under either Assumption 1 or Assumption 2. To show that $g_x$ is a PC mapping, we first note that, by the monotonicity of $f_x$, one has $f_x(z) \geq f_x(0)$ for $z \in \mathbb{R}^n_+$.

Under Assumption 1, we have $y_0 := f_x(0) = x \gg 0$, so that $(\forall z \in \mathbb{R}^n_+)\ f_x(z) \gg 0$. We now note that a selection of any of the scalar activation functions from List 1 for $\sigma$ satisfies $(\forall v \gg 0)\ \sigma(v) \gg 0$, so that $\sigma(y_0) = (\sigma \circ f_x)(0) = g_x(0) \gg 0$. Since $g_x$ is monotonic, one has $(\forall y \in \mathbb{R}^n_+)\ g_x(y) \geq g_x(0) \gg 0$, thus $g_x$ is a PC mapping.

On the other hand, if Assumption 2 is satisfied, we note that, in this case, $\sigma(0) \gg 0$, hence $(\forall z \in \mathbb{R}^n_+)\ g_x(z) = (\sigma \circ f_x)(z) \gg 0$, which implies $g_x$ is also a PC mapping in this case. ■

For convenience, the DEQ layers satisfying the assumptions in Lemma 4.2 are formally defined below.

**Definition 4.3.** Let $g_x \colon \mathbb{R}^n_+ \to \text{int}(\mathbb{R}^n_+)$ be a DEQ layer of the form in (1) for a given input $x$. If $g_x$ (including the input $x$) satisfies Assumption 1 in Lemma 4.2, then $g_x$ is called a pcDEQ-1 layer. If $g_x$ satisfies Assumption 2 in Lemma 4.2, then $g_x$ is called a pcDEQ-2 layer. By pcDEQ we mean a layer $g_x$ that is either a pcDEQ-1 layer or a pcDEQ-2 layer.

PcDEQ-1 and pcDEQ-2 layers are illustrated in Figure 1.

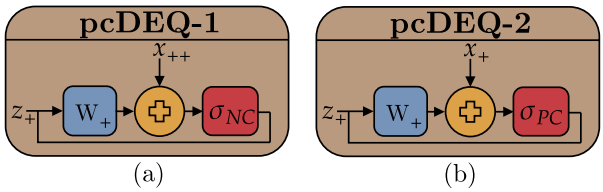


*Figure 1.* The visualization of the possible construction of pcDEQ layers. The symbols shown in the figures mean: $\mathbb{R}^{n\times n}_+ \ni W_+$ - nonnegative weights, $\mathbb{R}^n_+ \ni z_+$ - nonnegative vector of fixed point iteration, $\mathbb{R}^n_+ \ni x_+$ - nonnegative input to the layer, $\text{int}(\mathbb{R}^n_+) \ni x_{++}$ - positive input to the layer, $\sigma_{NC}$ - nonnegative concave activation function (List 1 in Remark 4.1) and $\sigma_{PC}$ - positive concave activation function (List 2 in Remark 4.1).

Lemma 4.2 asserts that pcDEQ layers are PC mappings for any allowed input $x$, which, in view of Proposition A.6, implies that they are also SI mappings. We can now use Definition A.2 to provide the form of the asymptotic mapping associated with a pcDEQ layer. The assertion of Proposition 4.4 follows from Proposition 11 in (Piotrowski et al., 2024), and, for completeness, we include the proof below.

**Proposition 4.4.** *For a given input $x$, let $g_{x,\infty}\colon \mathbb{R}^n_+ \to \mathbb{R}^n_+$ be the asymptotic mapping (in the sense of Definition A.2) of a pcDEQ layer $g_x\colon \mathbb{R}^n_+ \to \mathrm{int}(\mathbb{R}^n_+)$ satisfying Definition 4.3. Then $g_{x,\infty}(z) = 0$ for every $z \in \mathbb{R}^n_+$.*

*Proof.* The asymptotic mapping $g_{x,\infty}$ according to Definition A.2 is defined as follows

$$g_{x,\infty}(z) = \lim_{p\to\infty} \frac{1}{p} g_x(pz) = \lim_{p\to\infty} \frac{\sigma(Wpz + x)}{p}. \quad (6)$$

Applying L'Hôpital's rule to the activation function $\sigma$ with $p$ as an argument, we have

$$g_{x,\infty}(z) = \lim_{p\to\infty} \sigma'(Wpz + x)Wz. \quad (7)$$

For activation functions composed from any scalar activation functions in Remark 4.1, we have $\lim_{p\to\infty} \sigma'(Wpz + x) = 0$, hence also

$$\lim_{p\to\infty} \sigma'(Wpz + x)Wz = 0. \blacksquare \quad (8)$$

The following Corollary 4.5 follows directly from Proposition 4.4.

**Corollary 4.5.** *With notation in Proposition 4.4, let $\rho(g_{x,\infty}) \in \mathbb{R}_+$ be the spectral radius of $g_{x,\infty}$ in the sense of Definition A.3 for a given input $x$. Then $\rho(g_{x,\infty}) = 0$.*

Based on the previous results, Proposition 4.6 establishes the existence and uniqueness of the fixed point for the pcDEQ layers, in addition to the geometric convergence of the fixed point iteration.

**Proposition 4.6.** *Let $g_x\colon \mathbb{R}^n_+ \to \mathrm{int}(\mathbb{R}^n_+)$ be a pcDEQ layer satisfying Definition 4.3. Then $g_x$ has a unique fixed point $z^\star$ for every input $x$ satisfying the conditions in Definition 4.3. Moreover, the fixed point iteration of $g_x$ in (2) converges geometrically in the sense of Definition A.1 to $z^\star$ for any $z_1 \in \mathbb{R}^n_+$ and any input $x$.*

*Proof.* Choose any input $x$ satisfying the conditions in Definition 4.3. From Lemma 4.2, $g_x$ is a PC mapping. Thus, from Proposition A.6, $g_x$ is also an SI mapping. From Corollary 4.5, the nonlinear spectral radius of $g_x$ is $\rho(g_{x,\infty}) = 0$. If follows from Proposition A.4 that $g_x$ has a unique fixed point $z^\star$. Then, from Proposition A.7 it follows that the fixed point iteration of $g_x$ in (2) converges geometrically to $z^\star$ with a factor $c \in [0, 1)$ for any starting point $z_1 \in \mathbb{R}^n_+$. $\blacksquare$

## 5. Experiments

The experiments were carried out on three commonly known computer vision datasets: MNIST, SVHN, and CIFAR-10. The proposed pcDEQ[1] models were compared with competitive approaches for which the uniqueness of the fixed point is mathematically established, namely: monotone operator deep equilibrium models (monDEQ) (Winston & Kolter, 2020), neural ordinary differential equations (NODE) (Chen et al., 2018), and augmented neural differential equations (ANODE) (Dupont et al., 2019). The comparison was performed according to the results reported in (Dupont et al., 2019; Winston & Kolter, 2020). For pcDEQ models, the fixed point $z^\star$ is computed using the standard fixed point iteration in (2). The stopping criterion of fixed point iteration is based on the relative error, calculated as $\frac{||z_{k+1}-z_k||}{||z_{k+1}||} \leq \epsilon$, where $||\cdot||$ is a Frobenius norm and $\epsilon$ is a tolerance. In our experiments, the tolerance $\epsilon$ was set to $1e-4$.

The experiments were performed using the Google Colab platform with a NVIDIA Tesla T4 16GB GPU.

### 5.1. Architecture Setup

According to Definition 4.3, we consider two options to build pcDEQ layers, pcDEQ-1 (Figure 1 (a)) and pcDEQ-2 (Figure 1 (b)). For both options, the nonnegative weights $W_+$ are achieved by projecting the negative values to zeros after backpropagation. To construct pcDEQ-1, it is necessary to provide a positive input $x_{++}$ (see Figure 1 (a)), which is performed by applying elementwise the softplus activation function ($\sigma\colon \mathbb{R} \to \mathrm{int}(\mathbb{R}_+)$) before the pcDEQ-1 layer, as the activation functions in networks with pcDEQ-1 layers are functions from List 1 in Remark 4.1. In the case of the pcDEQ-2 model, the activation function is a PC activation function from List 2 in Remark 4.1. For pcDEQ-2, the nonnegativity of $x_+$ is achieved by applying the ReLU activation function before the DEQ layer.

Compared to monDEQ models, the construction of pcDEQ layers is much simpler and does not require a special layer implementation. MonDEQ layer has to be carefully parameterized by a set of two weights to satisfy the assumptions of strong monotonicity, which results in overparameterization and a more complicated implementation. Furthermore, computing convolutions in monDEQ models requires calculating fast Fourier transforms, which, as noted in (Winston & Kolter, 2020), are empirically 2-3 times slower than computing convolutions in a standard manner. On the other hand, in pcDEQ models, as mentioned previously, the only requirement is to use concave activation functions and constrain the weights to be nonnegative. Such a parameterization does not produce extra computational overhead compared to standard DEQ models.

In experiments, we consider three types of pcDEQ mod-

[1]The *Pytorch* source code of pcDEQs along with examples of use is available at the following link: `https://github.com/mateuszgabor/pcdeq`

els with four activation functions from Remark 4.1. The first type of network is based on using one linear pcDEQ layer. The second type of network is based on using one convolutional pcDEQ layer, and the last type is based on the use of three convolutional pcDEQ layers, between which the explicit downsampling layers occur. The architectural details are discussed in Appendix B, and the experimental hyperparameters for each network in Appendix C. Networks using a single linear pcDEQ layer have "L" suffix in their name. A similar scenario is for networks with a single convolutional pcDEQ layer, to network name suffix "SC" is added. For networks with three convolutional pcDEQ layers, the suffix "MC" is added to the network name. The suffix "MT" in monDEQs refers to the multi-tier architecture used in (Winston & Kolter, 2020).

### 5.2. Results

Tables 1, 2, and 3 show the results obtained by pcDEQ models with compared methods for MNIST, SVHN and CIFAR-10, respectively. As the results show, the architectures based on pcDEQ achieve competitive results compared to other implicit models. In each scenario, pcDEQ models were constructed with a smaller number of parameters compared to NODE, ANODE, and monDEQ. In the case of the results obtained on the MNIST dataset (Table 1), all pcDEQ configurations outperform the NODE, ANODE, and monDEQ approaches. For the results obtained on the SVHN dataset (Table 2), it can be seen that the highest accuracy is obtained by pcDEQ with ReLU6 activation functions and three convolutional layers. In the case of results obtained on the CIFAR-10 dataset, Table 3 shows that with a lower number of parameters, pcDEQ models can achieve similar or better results compared to NODE, ANODE, or monDEQ. Moreover, similar to previous works, we trained larger pcDEQ models with data augmentation on CIFAR-10. As can be seen, for this setup, pcDEQ with three pcDEQ convolutional layers and softsign activation functions achieves the highest accuracy among all pcDEQ configurations.

Figure 2 shows the training curves for the pcDEQ models with a single convolutional layer. For other cases, the figures are attached in Appendix D.

### 5.3. Convergence Analysis

Figure 3 shows the average number (among all batches in epoch) of fixed point iterations to compute a fixed point per epoch for the pcDEQ model with a single convolutional layer. From this figure, it can be seen that the convergence is very fast and accurate. For pcDEQ with sigmoid activation function, the fixed point iteration satisfies the stopping criterion with less than eight iterations. From Proposition 4.6, we know that the convergence of the fixed point iteration for the pcDEQ models is geometric. We could improve the convergence rate with vector extrapolation techniques. This fact is very interesting because such fast convergence is achieved without using any acceleration method. The interesting fact is that in standard DEQ and monDEQ models, the number of iterations of the used iterative method increases during training; in the proposed pcDEQ architectures, this effect does not occur. For other architecture configurations, the

*Table 1.* Test accuracies of pcDEQ models averaged over five runs on MNIST dataset compared with NODE, ANODE and monDEQ; † as reported in (Dupont et al., 2019); ‡ as reported in (Winston & Kolter, 2020).

| **Method** | **MNIST** | |
|---|---|---|
| | #Parameters | Accuracy [%] |
| NODE† | 84K | 96.4 |
| ANODE† | 84K | 98.2 |
| monDEQ-L‡ | 84K | 98.1 |
| monDEQ-SC‡ | 84K | 99.1 |
| monDEQ-MT‡ | 81K | 99.0 |
| pcDEQ-1-L-ReLU6 | 70K | 98.1 |
| pcDEQ-1-L-Tanh | 70K | 98.2 |
| pcDEQ-1-L-Softsign | 70K | 98.1 |
| pcDEQ-2-L-Sigmoid | 70K | 98.1 |
| pcDEQ-1-SC-ReLU6 | 69K | 99.2 |
| pcDEQ-1-SC-Tanh | 69K | 99.2 |
| pcDEQ-1-SC-Softsign | 69K | 99.1 |
| pcDEQ-2-SC-Sigmoid | 69K | 98.9 |
| pcDEQ-1-MC-ReLU6 | 41K | 99.3 |
| pcDEQ-1-MC-Tanh | 41K | 99.2 |
| pcDEQ-1-MC-Softsign | 41K | 99.2 |
| pcDEQ-2-MC-Sigmoid | 41K | 98.7 |

*Table 2.* Test accuracies of pcDEQ models averaged over five runs on SVHN dataset compared with NODE, ANODE and monDEQ; † as reported in (Dupont et al., 2019); ‡ as reported in (Winston & Kolter, 2020).

| **Method** | **SVHN** | |
|---|---|---|
| | #Parameters | Accuracy [%] |
| NODE† | 172K | 81.0 |
| ANODE† | 172K | 83.5 |
| monDEQ-SC‡ | 172K | 88.7 |
| monDEQ-MT‡ | 170K | 92.4 |
| pcDEQ-1-SC-ReLU6 | 165K | 88.0 |
| pcDEQ-1-SC-Tanh | 165K | 88.1 |
| pcDEQ-1-SC-Softsign | 165K | 88.4 |
| pcDEQ-2-SC-Sigmoid | 165K | 87.3 |
| pcDEQ-1-MC-ReLU6 | 131K | 93.0 |
| pcDEQ-1-MC-Tanh | 131K | 92.3 |
| pcDEQ-1-MC-Softsign | 131K | 92.3 |
| pcDEQ-2-MC-Sigmoid | 131K | 91.5 |

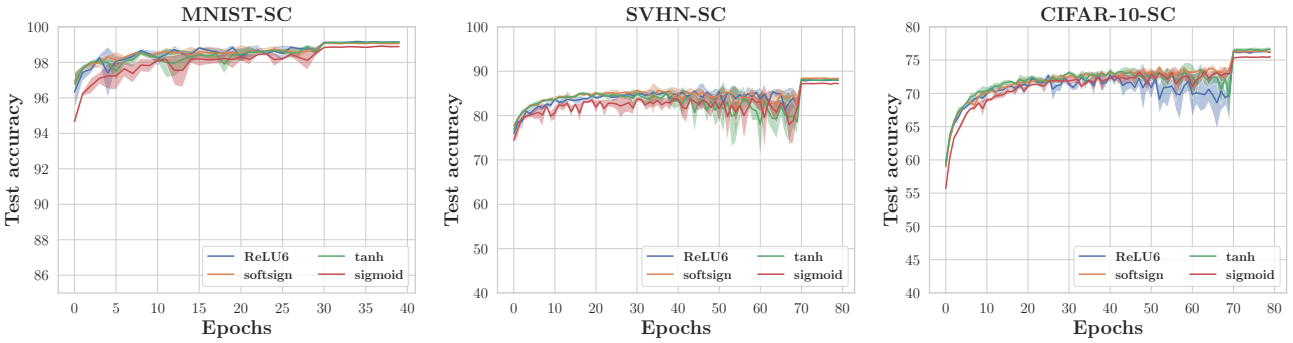


*Figure 2.* Test accuracies during training for the pcDEQ model with a single convolutional layer over five experiment runs.

*Table 3.* Test accuracies of pcDEQ models averaged over five runs on CIFAR-10 dataset compared with NODE, ANODE and monDEQ; † as reported in (Dupont et al., 2019); ‡ as reported in (Winston & Kolter, 2020); * with data augmentation.

| **Method** | **CIFAR-10** | |
|---|---|---|
| | #Parameters | Accuracy [%] |
| NODE† | 172K | 53.7 |
| NODE‡* | 1M | 59.9 |
| ANODE† | 172K | 60.6 |
| ANODE‡* | 1M | 73.4 |
| monDEQ-SC‡ | 172K | 74.0 |
| monDEQ-SC‡* | 854K | 82.0 |
| monDEQ-MT‡ | 170K | 72.0 |
| monDEQ-MT‡* | 1M | 89.0 |
| pcDEQ-1-SC-ReLU6 | 165K | 76.3 |
| pcDEQ-1-SC-Tanh | 165K | 76.6 |
| pcDEQ-1-SC-Softsign | 165K | 76.4 |
| pcDEQ-2-SC-Sigmoid | 165K | 75.5 |
| pcDEQ-1-MC-ReLU6 | 131K | 78.2 |
| pcDEQ-1-MC-ReLU6* | 661K | 89.2 |
| pcDEQ-1-MC-Tanh | 131K | 76.5 |
| pcDEQ-1-MC-Tanh* | 661K | 88.5 |
| pcDEQ-1-MC-Softsign | 131K | 77.1 |
| pcDEQ-1-MC-Softsign* | 661K | 89.0 |
| pcDEQ-2-MC-Sigmoid | 131K | 71.0 |
| pcDEQ-2-MC-Sigmoid* | 661K | 85.6 |

situation is similar to that described in this section, and the results are given in Appendix D.

### 5.4. Lipschitz Continuity

In this section, by considering the standard Euclidean metric space, we show empirically that the assumptions used to determine the uniqueness of the fixed point are weaker compared to the standard assumptions in convex analysis. In general, to determine the uniqueness of the fixed point based on the Banach fixed point theorem, we need information about a Lipschitz constant of the function, and, in the machine learning literature, we often consider the standard Euclidean space. In the case of neural networks, if the used activation functions are nonexpansive (such as those in Remark 4.1), then a Lipschitz constant $L$ w.r.t. the standard Euclidean space is upper bounded by the product of spectral norms of weight operators (Combettes & Pesquet, 2020) as follows:

$$L \leq \prod_{i=1}^{m} ||W_i||, \tag{9}$$

where $m$ is the number of weight operators and $||\cdot||$ is the spectral norm.

For the simple scenario, we investigated the pcDEQ model containing one linear pcDEQ layer ($m = 1$) with four activation functions. The average largest singular value per epoch for the entire training process is shown in Figure 4. It can be concluded that for the network with softsign, hyperbolic tangent, and sigmoid activation function, the Lipschitz constant in (9) is $L > 1$ for most of the training time. For such a case, the uniqueness of the fixed point cannot be determined from the Banach fixed point theorem, because the Lipschitz constant $L$ in (9) does not satisfy $L < 1$. In the case of a network with the ReLU6 activation function, the Lipschitz constant in (9) satisfies $L < 1$ for the entire training process. In such a case, the Banach fixed point theorem can be used to determine the uniqueness of the fixed point and the linear convergence to it by using fixed point iteration. However, there are no guarantees that even for ReLU6 the Lipschitz constant $L$ will always be less than one, for example, with different learning rates, initialization, architecture, etc. On the other hand, the necessary and sufficient conditions of the uniqueness of the fixed point of SI mappings are weaker compared to the Banach fixed point theory, because the uniqueness of the fixed point is independent of a Lipschitz constant and the unique fixed

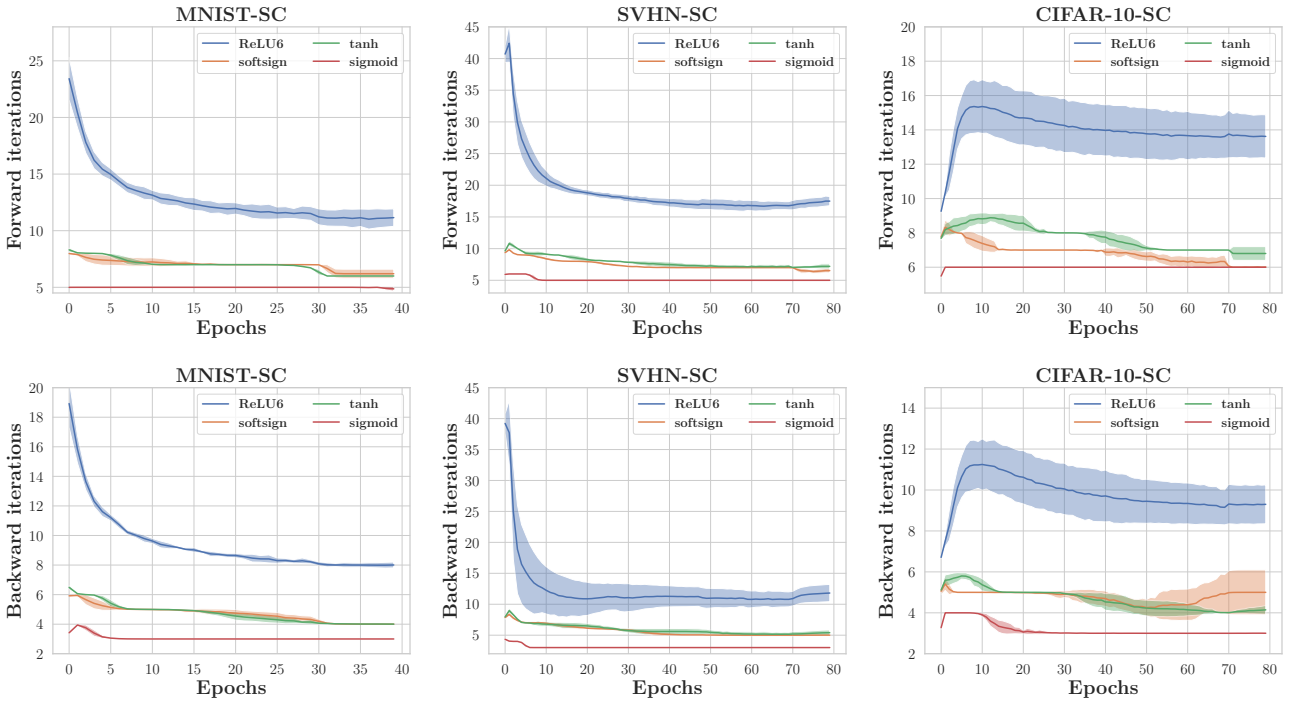


*Figure 3.* Number of fixed point iterations for computing the fixed point in forward and backward passes for the pcDEQ model with a single convolutional layer over five experiment runs.

point can still exist even if $L > 1$.

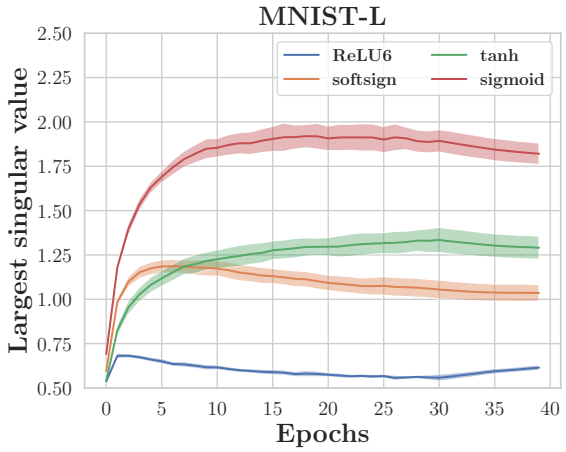


*Figure 4.* Largest singular value of pcDEQ linear layer over five experiment runs.

## 6. Limitations

The proposed approach may appear restrictive due to the non-negativity constraint on weights and the use of concave activation functions. However, these restrictions are essential to ensure theoretical guarantees, such as the uniqueness of the fixed point and the convergence of the standard fixed point iteration. It is important to note that existing approaches offering similar guarantees, such as monDEQ models, also impose constraints to ensure that the mappings are strongly monotone in the sense used in convex analysis. The advantage of our approach is that the imposed restrictions are simpler to implement in practice compared to those based on monotone operator theory.

## 7. Conclusions

In this study, we have proposed a new class of DEQ models with guarantees of the existence of a unique fixed point. The parametrization of pcDEQ is very simple, and it does not require any sophisticated training modifications compared to standard DEQ models. Moreover, the fixed point can be easily computed with the standard fixed point iteration, and the convergence is guaranteed to be geometric. The proposed pcDEQ models are based on the theory of SI mappings, which are widely used in the wireless literature. To the best of our knowledge, this is the first practical application of SI mappings theory in the deep learning literature.

This study provides the foundation for extending the proposed method to a larger and less restricted class of mappings with guarantees similar to SI mappings. As an example, one may aim to weaken further the assumptions used to construct pcDEQ models to obtain more versatile models, capable of incorporating a wider class of weight operators

and activation functions in DEQ layers. Such a direction seems feasible in view of the recent results on the existence and shape of the fixed point sets of (subhomogeneous) weakly standard interference (WSI) neural networks provided in (Piotrowski et al., 2024). Another promising future research in the direction proposed in this paper should focus on providing even stronger guarantees of convergence rate compared with the geometric convergence of the fixed point iteration of pcDEQ models established in this paper. Indeed, the empirical convergence analysis provided in Section 5.3 suggests that the actual rate of convergence may actually be linear.

## Acknowledgments

This work is supported by the National Center of Science under Preludium 22 project No. UMO-2023/49/N/ST6/02697 (Mateusz Gabor), National Centre for Research and Development of Poland (NCBR) under grant EIG CONCERT-JAPAN/04/2021, and by the Federal Ministry of Education and Research of Germany under grant 01DR21009 and the programme "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification numbers: 16KISK020K and 16KISK030.

## Impact Statement

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

## References

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

Bai, S., Koltun, V., and Kolter, J. Z. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33:5238–5250, 2020.

Bai, S., Geng, Z., Savani, Y., and Kolter, J. Z. Deep equilibrium optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 620–630, 2022.

Baker, J., Wang, Q., Hauck, C. D., and Wang, B. Implicit graph neural networks: a monotone operator viewpoint. In *International Conference on Machine Learning*, pp. 1521–1548. PMLR, 2023.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2th edition edition, 2017.

Bauschke, H. H., Combettes, P. L., Bauschke, H. H., and Combettes, P. L. *Correction to: Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

Cavalcante, R. L., Shen, Y., and Stańczak, S. Elementary properties of positive concave mappings with applications to network planning and optimization. *IEEE Transactions on Signal Processing*, 64(7):1774–1783, 2016.

Cavalcante, R. L., Liao, Q., and Stańczak, S. Connections between spectral properties of asymptotic mappings and solutions to wireless network problems. *IEEE Transactions on Signal Processing*, 67(10):2747–2760, 2019.

Chen, Q., Wang, Y., Geng, Z., Wang, Y., Yang, J., and Lin, Z. Equilibrium image denoising with implicit differentiation. *IEEE Transactions on Image Processing*, 32:1868–1881, 2023.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Combettes, P. L. and Pesquet, J.-C. Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM Journal on Mathematics of Data Science*, 2(2):529–557, 2020.

Dupont, E., Doucet, A., and Teh, Y. W. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019.

El Ghaoui, L., Gu, F., Travacca, B., Askari, A., and Tsai, A. Implicit deep learning. *SIAM Journal on Mathematics of Data Science*, 3(3):930–958, 2021.

Ertenli, C. U., Akbas, E., and Cinbis, R. G. Streaming multiscale deep equilibrium models. In *European Conference on Computer Vision*, pp. 189–205. Springer, 2022.

Geng, Z., Zhang, X.-Y., Bai, S., Wang, Y., and Lin, Z. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

Geng, Z., Pokle, A., and Kolter, J. Z. One-step diffusion distillation via deep equilibrium models. 2023.

Gilton, D., Ongie, G., and Willett, R. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021.

Gkillas, A., Ampeliotis, D., and Berberidis, K. Connections between deep equilibrium and sparse representation models with application to hyperspectral image denoising. *IEEE Transactions on Image Processing*, 32:1513–1528, 2023a.

Gkillas, A., Ampeliotis, D., and Berberidis, K. Deep equilibrium models meet federated learning. *arXiv preprint arXiv:2305.18646*, 2023b.

Kolter, Z., Duvenaud, D., and Johnson, M. Deep implicit layers-neural odes, deep equilibirum models, and beyond. *URL http://implicit-layers-tutorial. org/(visited on 2022-03-25)*, 2020.

Koyama, Y., Murata, N., Uhlich, S., Fabbro, G., Takahashi, S., and Mitsufuji, Y. Music source separation with deep equilibrium models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 296–300. IEEE, 2022.

Krause, U. *Positive dynamical systems in discrete time: theory, models, and applications*, volume 62. Walter de Gruyter GmbH & Co KG, 2015.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lemmens, B. and Nussbaum, R. *Nonlinear Perron-Frobenius Theory*, volume 189. Cambridge University Press, 2012.

Lins, B. A unified approach to nonlinear perron-frobenius theory. *Linear Algebra and its Applications*, 2023.

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

Miretti, L., Cavalcante, R. L., and Björnson, E. Ul-dl duality for cell-free networks under per-ap power and information constraints. In *ICC 2023-IEEE International Conference on Communications*, pp. 5017–5023. IEEE, 2023a.

Miretti, L., Cavalcante, R. L., and Stańczak, S. Fixed-point methods for long-term power control and beamforming design in large-scale mimo. 2023b. submitted, available as arXiv preprint arXiv:2312.02080.

Ortega, J. M. and Rheinboldt, W. C. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Oshime, Y. Perron-frobenius problem for weakly sublinear maps in a euclidean positive orthant. *Japan journal of industrial and applied mathematics*, 9:313–350, 1992.

Piotrowski, T. and Cavalcante, R. L. G. The fixed point iteration of positive concave mappings converges geometrically if a fixed point exists: implications to wireless systems. *IEEE Transactions on Signal Processing*, 70: 4697–4710, 2022.

Piotrowski, T. J., Cavalcante, R. L. G., and Gabor, M. Fixed points of nonnegative neural networks. *Journal of Machine Learning Research*, 25(139):1–40, 2024.

Pokle, A., Geng, Z., and Kolter, J. Z. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems*, 35:37975–37990, 2022.

Revay, M., Wang, R., and Manchester, I. R. Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*, 2020.

Ryu, E. K. and Boyd, S. Primer on monotone operator methods. *Appl. comput. math*, 15(1):3–43, 2016.

Salimans, T. and Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Schubert, M. and Boche, H. *Interference calculus: A general framework for interference management and network utility optimization*, volume 7. Springer Science & Business Media, 2011.

Shindoh, S. The structures of sinr regions for standard interference mappings. In *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1280–1285. IEEE, 2019.

Shindoh, S. Some properties of sinr regions for standard interference mappings. *SICE Journal of Control, Measurement, and System Integration*, 13(3):50–56, 2020.

Stanczak, S., Wiczanowski, M., and Boche, H. *Fundamentals of resource allocation in wireless networks: theory and algorithms*, volume 3. Springer Science & Business Media, 2009.

Tsuchida, R. and Ong, C. S. Deep equilibrium models as estimators for continuous latent variables. In *International Conference on Artificial Intelligence and Statistics*, pp. 1646–1671. PMLR, 2023.

Wang, S., Teng, Y., and Wang, L. Deep equilibrium object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6296–6306, 2023.

Wei, C. and Kolter, J. Z. Certified robustness for deep equilibrium models via interval bound propagation. In *International Conference on Learning Representations*, 2021.

Winston, E. and Kolter, J. Z. Monotone operator equilibrium networks. *Advances in neural information processing systems*, 33:10718–10728, 2020.

Xie, X., Wang, Q., Ling, Z., Li, X., Liu, G., and Lin, Z. Optimization induced equilibrium networks: An explicit optimization perspective for understanding equilibrium models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3604–3616, 2022.

Yamada, I., Yukawa, M., and Yamagishi, M. Minimizing the moreau envelope of nonsmooth convex functions over the fixed point set of certain quasi-nonexpansive mappings. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 345–390, 2011.

Yates, R. D. A framework for uplink power control in cellular radio systems. *IEEE Journal on selected areas in communications*, 13(7):1341–1347, 1995.

You, L. and Yuan, D. A note on decoding order in user grouping and power optimization for multi-cell noma with load coupling. *IEEE Transactions on Wireless Communications*, 20(1):495–505, 2020.

Zhang, S., Zhu, L., and Gao, Y. An efficient deep equilibrium model for medical image segmentation. *Computers in Biology and Medicine*, 148:105831, 2022.

Zhao, Y., Zheng, S., and Yuan, X. Deep equilibrium models for snapshot compressive imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 3642–3650, 2023.

Zheng, Z., Zhou, Y., and Zhou, H. Deep equilibrium non-autoregressive sequence learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 11763–11781, 2023.

Zou, Z., Liu, J., Wohlberg, B., and Kamilov, U. S. Deep equilibrium learning of explicit regularization functionals for imaging inverse problems. *IEEE Open Journal of Signal Processing*, 2023.

## A. Known Results

**Definition A.1.** (Ortega & Rheinboldt, 2000)[Chapter 9] Let $(x_k : k \in \mathbb{N})$ be a sequence in $\mathbb{R}^n$, then $(x_k : k \in \mathbb{N})$ converges geometrically to $x^\star \in \mathbb{R}^n$ with a rate $c \in [0, 1)$ and a constant $\gamma > 0$ if

$$\forall k \in \mathbb{N} \quad ||x_{k+1} - x^\star|| \leq \gamma c^k, \tag{10}$$

where $|| \cdot ||$ is a given norm.

**Definition A.2.** (Cavalcante et al., 2019; Oshime, 1992) Let $f \colon \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ be an SI mapping in the sense of Definition 3.2. The asymptotic mapping associated with $f$ is the mapping defined by

$$f_\infty : \mathbb{R}^n_+ \to \mathbb{R}^n_+ : z \mapsto \lim_{p \to \infty} \frac{1}{p} f(pz). \tag{11}$$

We recall that the above limit always exists and that the resulting asymptotic mapping $f_\infty$ is positively homogeneous; i.e., $(\forall \alpha > 0)\ (\forall z \in \mathbb{R}^n_+)\ f_\infty(\alpha x) = \alpha f_\infty(z)$.

**Definition A.3.** The (nonlinear) spectral radius of an SI mapping is defined as the largest eigenvalue of the corresponding asymptotic mapping (Cavalcante et al., 2019; Oshime, 1992), and it is given by

$$p(f_\infty) := \max\{\lambda \in \mathbb{R}_+ \mid \exists z \in \mathbb{R}^n_+ \setminus \{0\}\ s.t.\ f_\infty(z) = \lambda z\} \in \mathbb{R}_+. \tag{12}$$

**Proposition A.4.** *(Cavalcante et al., 2019) Let $f \colon \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ be an SI mapping. Then Fix$(f) \neq \emptyset$ if and only if $p(f_\infty) < 1$. Furthermore, if a fixed point exists, then it is positive and unique.*

**Corollary A.5.** *(Cavalcante et al., 2016) If $f \colon \mathbb{R}^n_+ \to \mathbb{R}^n_+$ is an NC mapping or $f \colon \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ is a PC mapping in the sense of Definition 3.4, then $f$ is monotonic.*

**Proposition A.6.** *(Cavalcante et al., 2016; 2019) If $f : \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ is a PC mapping, then $f$ is an SI mapping.*

**Proposition A.7.** *(Piotrowski & Cavalcante, 2022) Let $f \colon \mathbb{R}^n_+ \to \operatorname{int}(\mathbb{R}^n_+)$ be a PC mapping with a fixed point $x^\star \in \operatorname{int}(\mathbb{R}^n_+)$. Then, for any $x_1 \in \mathbb{R}^n_+$, the fixed point iteration of $f$ converges geometrically to $x^\star$ with a factor $c \in [0, 1)$ w.r.t. any metric induced by a norm in $\mathbb{R}^n$.*

## B. Architecture Details

As noted in Section 5, we consider three types of architectures of pcDEQ models. Based on the results of previous DEQ papers (Bai et al., 2019; 2020), using some regularization techniques can prevent overfitting and improve the final results. Therefore, similar to standard DEQ models (Bai et al., 2019) for each pcDEQ layer, we apply weight normalization (Salimans & Kingma, 2016). Similarly to the official tutorial on implicit models (Kolter et al., 2020), we also used batch normalization before and after the DEQ layer, which allows improving the final results. In Figures 5, 6, 7, the meaning of the blocks is as follows:

- Linear — linear layer,
- BN — batch normalization layer,
- pcDEQ-1 — pcDEQ-1 layer in Definition 4.3,
- pcDEQ-2 — pcDEQ-2 layer in Definition 4.3,
- Conv2D — 2D convolutional layer,
- Conv2D, s=2 — 2D convolutional downsampling layer with stride equal to 2,
- BN + Softplus — batch normalization layer followed by a softplus activation function,
- BN + ReLU — batch normalization followed by ReLU activation functions,
- MaxPool + BN — max pooling layer followed by batch normalization,
- AvgPool — average pooling layer.

### B.1. Architecture with Single pcDEQ Linear Layer

Architectures of pcDEQ models with a single linear pcDEQ layer for two variants of pcDEQ layers in the sense of Definition 4.3 are presented in Figure 5. As in standard DEQ models, the first and last linear layer is an explicit layer.

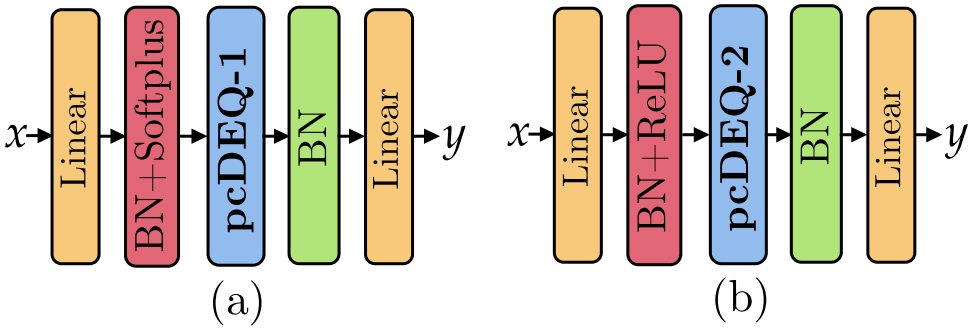


*Figure 5.* Architectures of pcDEQ models with single linear pcDEQ layer. Subfigure (a) presents architecture with pcDEQ-1 layer and (b) with pcDEQ-2 layer.

### B.2. Architecture with Single pcDEQ Convolutional Layer

Architectures of pcDEQ models with a single convolutional pcDEQ layer are similar to those with linear layers in the previous subsection. Figure 6 shows the architectures used in the experiments. The max. pooling layers have a kernel of size $3 \times 3$ with padding $p = 1$ and stride $s = 1$. The avg. pooling layer has kernel of size $8 \times 8$ with padding $p = 0$ and stride $s = 8$.

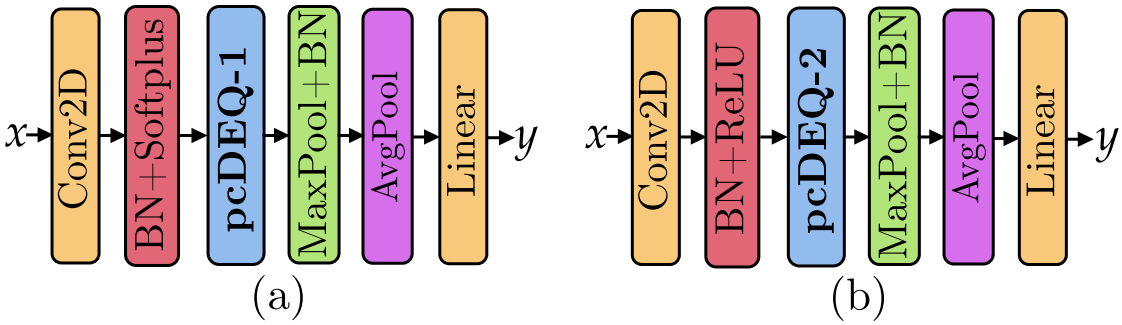


*Figure 6.* Architecture of pcDEQ models with single convolutional pcDEQ layer. Subfigure (a) presents architecture with pcDEQ-1 layer and (b) with pcDEQ-2 layer.

### B.3. Architecture with Multiple pcDEQ Convolutional Layers

In practice, larger DEQ models with multiple convolutional layers are implemented as single layer fusion of multiple scales (Bai et al., 2020), between which the upsampling and downsampling are performed. The same idea was applied to monDEQ models in the mult-tier architecture. This is a nice engineering idea, but it is complicated in practice. Moreover, computing the fusion of fixed points (for each scale) simultaneously is difficult to analyze and can involve an increased computational cost. In this work, we take another approach that combines implicit DEQ layers with explicit downsampling layers between them. A similar approach was used in (Xie et al., 2022), which investigated the idea of having multiple implicit layers instead of one. It should be noted that the explicit downsampling layers are unconstrained, such as the first and last layers. The proposed architecture is shown in Figure 7. The max. pooling layers have a kernel of size $3 \times 3$ with padding $p = 1$ and stride $s = 1$. The avg. pooling layer has kernel of size $4 \times 4$ with padding $p = 0$ and stride $s = 4$.

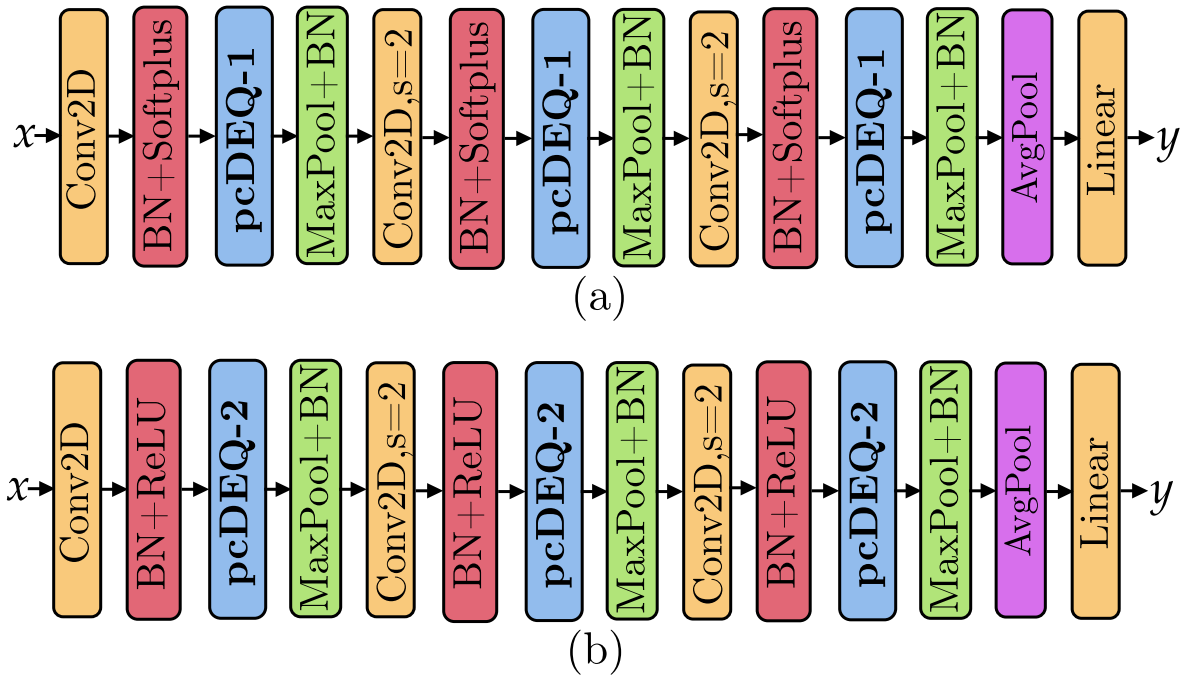


*Figure 7.* Architecture of pcDEQ models with three convolutional pcDEQ layer. Subfigure (a) presents architecture with pcDEQ-1 layers and (b) with pcDEQ-2 layers.

## C. Experimental Details and Hyperparameters

In our experiments, we used three commonly known computer vision datasets: MNIST, SVHN, and CIFAR-10. MNIST dataset consists 70,000 grayscale handwritten digit images. SVHN dataset consists of 99,289 RGB digit images from house numbers. CIFAR-10 consists of 60,000 RGB images of 10 classes. For the MNIST dataset, the images have dimensions of $28 \times 28$ pixels, and for the SVHN and CIFAR-10 $32 \times 32$ pixels. The statistics of the datasets are shown in Table 4.

*Table 4.* Dataset statistics.

| **Dataset** | **#Train examples** | **#Test examples** |
|---|---|---|
| MNIST | 60,000 | 10,000 |
| SVHN | 73,257 | 26,032 |
| CIFAR-10 | 50,000 | 10,000 |

As mentioned in the paper, to compute fixed point in pcDEQ layers, the standard fixed point iteration was used. We especially do not use any more sophisticated iterative methods such as Anderson acceleration or Broyden's method, because for such methods the guarantees of convergence to fixed point for PC mappings have not been proved in the literature. For all networks, the AdamW (Loshchilov & Hutter, 2017) optimizer was used with a batch size of 64. All other hyperparameters for each dataset and architecture are shown in Tables 5, 6 and 7. The expeeriments with pcDEQs were run five times.

*Table 5.* MNIST hyperparameters.

| Method | Number of channels | Epochs | LR | LR decay steps | LR decay factor | WD |
|---|---|---|---|---|---|---|
| pcDEQ-1-L-ReLU6 | 80 | 40 | 0.001 | 30 | 0.1 | 0.02 |
| pcDEQ-1-L-Tanh | 80 | 40 | 0.001 | 30 | 0.1 | 0.02 |
| pcDEQ-1-L-Softsign | 80 | 40 | 0.001 | 30 | 0.1 | 0.02 |
| pcDEQ-2-L-Sigmoid | 80 | 40 | 0.001 | 30 | 0.1 | 0.02 |
| pcDEQ-1-SC-ReLU6 | 82 | 40 | 0.0007 | 30 | 0.1 | 0.02 |
| pcDEQ-1-SC-Tanh | 82 | 40 | 0.0007 | 30 | 0.1 | 0.02 |
| pcDEQ-1-SC-Softsign | 82 | 40 | 0.0007 | 30 | 0.1 | 0.02 |
| pcDEQ-2-SC-Sigmoid | 82 | 40 | 0.0002 | 30 | 0.1 | 0.02 |
| pcDEQ-1-MC-ReLU6 | 12,24,48 | 40 | 0.0005 | 30 | 0.1 | 0.015 |
| pcDEQ-1-MC-Tanh | 12,24,48 | 40 | 0.0005 | 30 | 0.1 | 0.015 |
| pcDEQ-1-MC-Softsign | 12,24,48 | 40 | 0.0005 | 30 | 0.1 | 0.015 |
| pcDEQ-2-MC-Sigmoid | 12,24,48 | 40 | 0.0002 | 30 | 0.1 | 0.015 |

*Table 6.* SVHN hyperparameters.

| Method | Number of channels | Epochs | LR | LR decay steps | LR decay factor | WD |
|---|---|---|---|---|---|---|
| pcDEQ-1-SC-ReLU6 | 125 | 80 | 0.0007 | 70 | 0.1 | 0.02 |
| pcDEQ-1-SC-Tanh | 125 | 80 | 0.0007 | 70 | 0.1 | 0.02 |
| pcDEQ-1-SC-Softsign | 125 | 80 | 0.0007 | 70 | 0.1 | 0.02 |
| pcDEQ-2-SC-Sigmoid | 125 | 80 | 0.0005 | 70 | 0.1 | 0.02 |
| pcDEQ-1-MC-ReLU6 | 20,50,80 | 50 | 0.0005 | 40 | 0.1 | 0.015 |
| pcDEQ-1-MC-Tanh | 20,50,80 | 50 | 0.0005 | 40 | 0.1 | 0.015 |
| pcDEQ-1-MC-Softsign | 20,50,80 | 50 | 0.0005 | 40 | 0.1 | 0.015 |
| pcDEQ-2-MC-Sigmoid | 20,50,80 | 50 | 0.0002 | 40 | 0.1 | 0.015 |

*Table 7.* CIFAR-10 hyperparameters.

| Method | Number of channels | Epochs | LR | LR decay steps | LR decay factor | WD |
|---|---|---|---|---|---|---|
| pcDEQ-1-SC-ReLU6 | 125 | 80 | 0.0005 | 70 | 0.1 | 0.02 |
| pcDEQ-1-SC-Tanh | 125 | 80 | 0.0005 | 70 | 0.1 | 0.02 |
| pcDEQ-1-SC-Softsign | 125 | 80 | 0.0005 | 70 | 0.1 | 0.02 |
| pcDEQ-2-SC-Sigmoid | 125 | 80 | 0.0002 | 70 | 0.1 | 0.02 |
| pcDEQ-1-MC-ReLU6 | 20,50,80 | 50 | 0.0007 | 40 | 0.1 | 0.015 |
| pcDEQ-1-MC-Tanh | 20,50,80 | 50 | 0.0007 | 40 | 0.1 | 0.015 |
| pcDEQ-1-MC-Softsign | 20,50,80 | 50 | 0.0007 | 40 | 0.1 | 0.015 |
| pcDEQ-2-MC-Sigmoid | 20,50,80 | 50 | 0.0002 | 40 | 0.1 | 0.015 |
| pcDEQ-1-MC-ReLU6* | 100,120,140 | 120 | 0.0007 | 100 | 0.1 | 0.02 |
| pcDEQ-1-MC-Tanh* | 100,120,140 | 120 | 0.0007 | 100 | 0.1 | 0.02 |
| pcDEQ-1-MC-Softsign* | 100,120,140 | 120 | 0.0007 | 100 | 0.1 | 0.02 |
| pcDEQ-2-MC-Sigmoid* | 100,120,140 | 120 | 0.0002 | 100 | 0.1 | 0.02 |

# D. Additional Figures

## D.1. MNIST

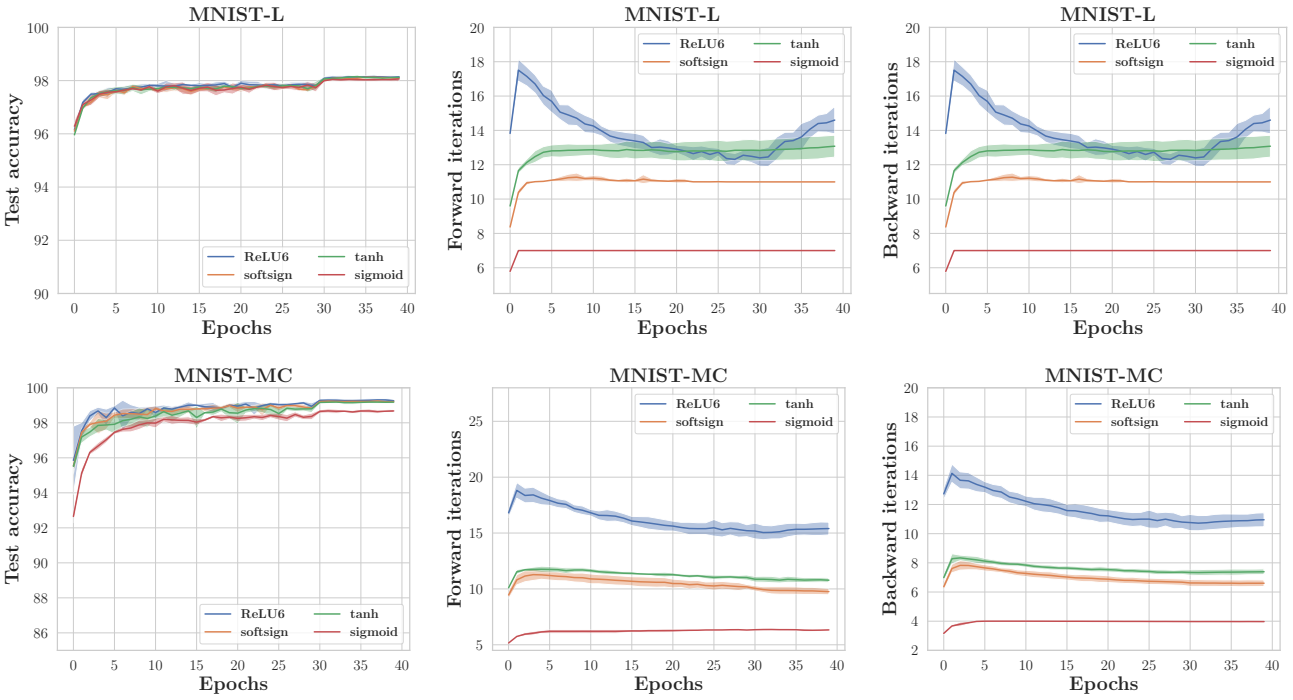


*Figure 8.* Test accuracies, number of fixed point iterations in forward and backward passes during training for pcDEQ models with single linear pcDEQ layer and three convolutional pcDEQ layers (average of forward and backward passes of three pcDEQ layers) on MNIST dataset over five experiment runs.

## D.2. SVHN

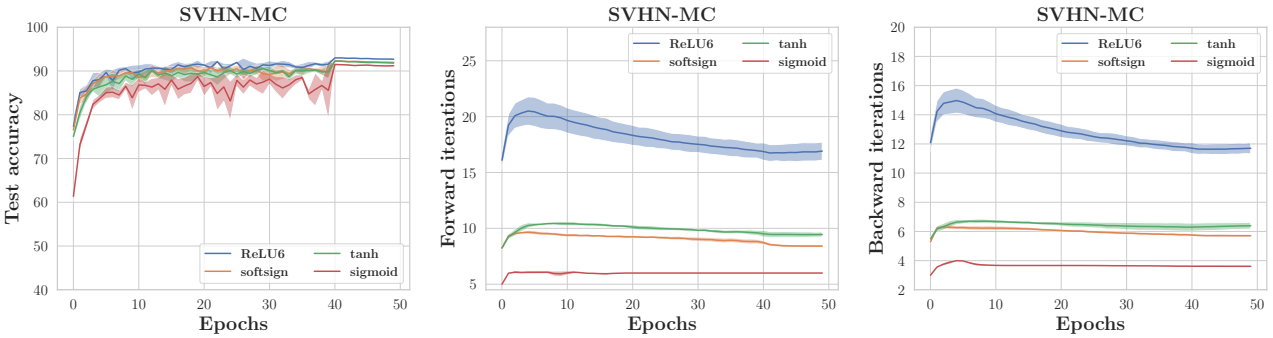


*Figure 9.* Test accuracies, number of fixed point iterations in forward and backward passes (average of three pcDEQ layers) during training for pcDEQ models with multiple convolutional pcDEQ layers on SVHN dataset over five experiment runs.

## D.3. CIFAR-10

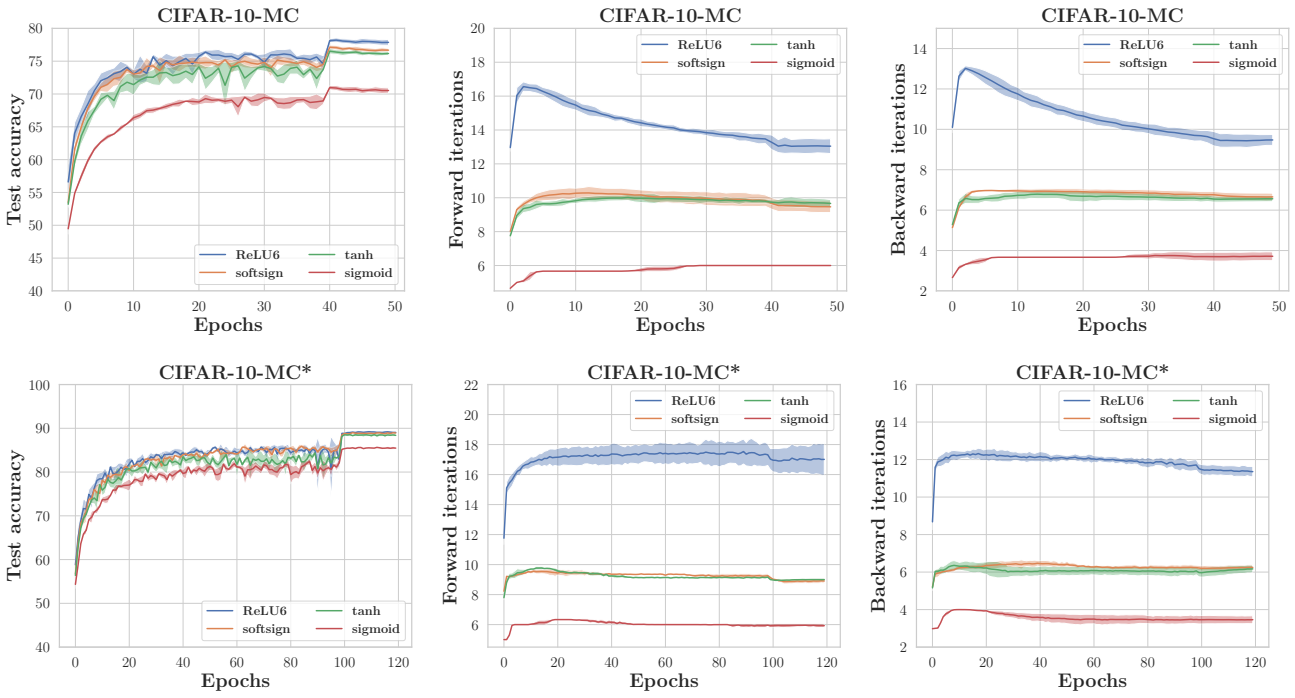


*Figure 10.* Test accuracies, number of fixed point iterations in forward and backward passes (average of three pcDEQ layers) during training for pcDEQ models with three convolutional pcDEQ layers with and without data augmentation on CIFAR-10 dataset over five experiment runs.