Subhomogeneous Deep Equilibrium Models

Pietro Sittoni [1] **Francesco Tudisco** [1 2]

Abstract

Implicit-depth neural networks have grown as powerful alternatives to traditional networks in various applications in recent years. However, these models often lack guarantees of existence and uniqueness, raising stability, performance, and reproducibility issues. In this paper, we present a new analysis of the existence and uniqueness of fixed points for implicit-depth neural networks based on the concept of subhomogeneous operators and the nonlinear Perron-Frobenius theory. Compared to previous similar analyses, our theory allows for weaker assumptions on the parameter matrices, thus yielding a more flexible framework for well-defined implicit networks. We illustrate the performance of the resulting subhomogeneous networks on feedforward, convolutional, and graph neural network examples.

1. Introduction

Implicit-depth Neural Networks (NNs) have emerged as powerful tools in deep learning. Rather than using a sequence of nested layers, these models define feature embeddings as the solution to specific nonlinear equations. Two popular examples of implicit-depth networks are Neural ODEs (Chen et al., 2018), whose output is calculated by solving an ordinary differential equation, and Deep Equilibrium (DEQ) Models (Bai et al., 2019), which define the output in terms of a fixed-point equation. These approaches have a number of advantages as (a) they have been shown to match or even exceed the performance of traditional NNs on several tasks, including time series and sequence modeling (Bai et al., 2019; Rusch & Mishra, 2021), and (b) memory-wise, they are more efficient than traditional NNs as backpropagation is done analytically and does not require storage of internal weights, allowing to handle deep NN architectures more efficiently.

DEQ models can be viewed as infinite-depth feed-forward neural networks, with weight tying, i.e. where the same transformation is used on each layer (Dabre & Fujita, 2019; Dehghani et al., 2019). Indeed, in this type of model, the evaluation of the network is executed by solving a fixed-point equation $z = f_\theta(z; x)$ which can be thought of as the limit for the number of layers $n \to \infty$ of an n-deep network $z^{(n)} = f_\theta(z^{(n-1)}; x)$. As the network is now implicitly defined as the solution of the equation $f_\theta(z; x) - z = 0$, one can use the Implicit Function Theorem to compute and propagate the gradients, thus reducing the memory cost with respect to standard backpropagation (Bai et al., 2019; 2020).

Despite their potential advantages, not all DEQ models are well-defined. In fact, one of the main open questions for DEQ architectures is whether any fixed point actually exists and whether this is unique. Lack of uniqueness, in particular, can be a problem as for a given data x and fixed pre-trained weights θ, the resulting fixed-point embedding z may change, raising stability, performance, and reproducibility issues. These potential drawbacks are often ignored or addressed only via empirical arguments based on experimental evidence that deep networks work well in practice. The most prominent line of analysis of uniqueness for deep equilibrium fixed points is based on monotone operator theory (Winston & Kolter, 2020; Ryu & Boyd, 2016). While elegant and efficient, monotone operator DEQs (MonDEQs) require special parametrizations of the layer weights to guarantee the DEQ model f_θ is operator monotone.

In this work, we present a new analysis of existence and uniqueness of DEQ fixed points based on positive, subhomogeneous operators. Using the Thomson projective metric and techniques from nonlinear Perron–Frobenius theory (Lemmens & Nussbaum, 2012; Gautier et al., 2019), we provide a new theorem showing that a broad class of operators f_θ admits unique fixed points. In particular, we show that our uniqueness theorem holds for several example architectures of the form $f_\theta(z; x) = \sigma(Wz) + \mathrm{MLP}(x)$, provided the activation function is subhomogeneous, which we show is the case for a variety of commonly used activations.

This existence and uniqueness result allow us to design stable DEQ models under much weaker assumptions than

[1]School of Mathematics, Gran Sasso Science Institute, L'Aquila, Italy [2]School of Mathematics and Maxwell Institute, University of Edinburgh, Edinburgh, UK. Correspondence to: Pietro Sittoni <pietro.sittoni@gssi.it>, Francesco Tudisco <f.tudisco@ed.ac.uk>.

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

available literature, avoiding any restriction on the learnable weight and using a large class of subhomogeneous activation functions.

Our theoretical findings are complemented by several experimental evaluations where we compare simple fully-connected and convolutional DEQ architectures based on monotone operators with the newly introduced subhomogeneous deep equilibrium model (SubDEQ) on benchmark image classification tasks.

2. Related work

The classical approach to layer design in deep learning is based on explicitly defined function compositions and the corresponding computational graph. In contrast, implicit-depth approaches do not explicitly define the computational graph and define the model implicitly as the solution of a specific equation. The computational graph needs then to be extracted from the implicit formulation. NeuralODE is a popular example where the model is defined as the solution of an ordinary differential equation and backpropagation can be implemented via the adjoint method (Chen et al., 2018). Another example is the DEQ, where the model is defined as the fixed point of a nonlinear function and backpropagation can be done using the implicit function theorem (Bai et al., 2019). However, DEQ may not be well-posed, as fixed points of nonlinear mappings may not exist or may not be unique. Several works in the literature propose variations of DEQ architecture with criteria for certified well-posedness using different methods, including monDEQ (Winston & Kolter, 2020) based on monotone operator theory, applying contraction theory (Jafarpour et al., 2021), using an over-parametrized DEQ with a condition on the initial equilibrium point, or exploiting linear and nonlinear Perron-Frobenius theory on graph neural networks and autoencoders, (Gu et al., 2020; El Ghaoui et al., 2021; Piotrowski et al., 2021). Just like standard explicit networks, DEQ models are vulnerable to adversarial input perturbations (Yang et al., 2022) and well-posed parameterizations may improve the robustness of the model. For example, using semialgebraic representation of monDEQ, Chen et al. (2021) showed that monDEQ are more robust to l^2 perturbation. Similarly, Wei & Kolter (2022) showed that unlike generic DEQs, monDEQ can achieve comparable l^∞ certified robustness to similarly-sized fully explicit networks, via interval bound propagation. Finally, we highlight that models based on DEQ architectures have been successfully employed in a wide range of specific applications. The first competitive performance was shown in the sequence modeling domain (Bai et al., 2019; 2020). Soon after that, efficient DEQ models have been designed for inverse problems in imaging (Gilton et al., 2021; Zhao et al., 2022), image denoising (Chen et al., 2023), optical flow estimation (Bai et al., 2022), landmark detection (Micaelli et al., 2023), semantic segmentation (Bai et al., 2020), and generative modeling using a diffusion-based approach (Pokle et al., 2022).

3. Subhomogeneous operators

The fundamental brick of our analysis is the notion of subhomogeneous operator. Recall that an operator $F : \mathbb{R}^n \to \mathbb{R}^n$ is (positively) μ-homogeneous, for some $\mu > 0$ if $F(\alpha z) = \alpha^\mu F(z)$ for all positive coefficients $\alpha > 0$ and all entrywise positive vectors $z > 0$. When F is differentiable, Euler's theorem for homogeneous functions provides an elegant equivalent characterization of homogeneous operators as the set of F such that the identity $F'(z)z = \mu F(z)$ holds for all z, where $F'(z)$ denotes the Jacobian of F in z. The proposed notion of subhomogeneous operators generalizes the concept of homogeneous mappings, starting from this second characterization.

Definition 3.1 (Subhomogeneous operator)**.** Let $F\colon \mathbb{R}^n \to \mathbb{R}^m$ be a Lipschitz mapping. For a coefficient $\mu > 0$ and $\Omega \subseteq \mathrm{dom}_+(F) := \{z \in \mathbb{R}^n \mid F(z) \geq 0\}$, we say that F is μ-subhomogeneus in Ω, briefly $F \in \mathrm{subhom}_\mu(\Omega)$, if

$$|Mz| \leq \mu\, F(z), \tag{1}$$

for all $z \in \Omega$ and all $M \in \partial F(z)$, where $\partial F(z)$ denotes Clarke's generalized Jacobian of F at the point z, and where all the inequalities, as well as the absolute value, are meant entrywise. Similarly, we say that F is strongly μ-subhomogeneus in Ω, briefly $F \in \text{s-subhom}_\mu(\Omega)$, if

$$|M|\,|z| \leq \mu\, F(z), \tag{2}$$

for all $z \in \Omega$ and all $M \in \partial F(z)$.

Subhomogeneity generalizes both homogeneity and strong-subhomogeneity. In fact, all μ-homogeneous differentiable operators are μ-subhomogeneous due to Euler's theorem and, as $|Mz| \leq |M||z|$ for all z, we immediately see that every strongly-subhomogeneus operator is subhomogeneus. However, homogeneity does not necessarily imply strong-subhomogeneity. This is shown for instance in Example 3.2 below. On the other hand, we will see that strong subhomogeneity is preserved under composition while subhomogeneity is not, and this additional property will be useful to establish uniqueness results for brother families of deep equilibrium architectures.

Example 3.2. Let $F\colon \mathbb{R}^2 \smallsetminus \{[0,0]\} \to \mathbb{R}^2$, be defined as $F(z) = F(x,y) = [\frac{x^2 y}{x^2+y^2}, 0]$. Clearly F is 1-homogeneus, with Jacobian given by

$$F'(z) = \begin{bmatrix} \frac{2xy^3}{(x^2+y^2)^2} & \frac{x^2(x^2-y^2)}{(x^2+y^2)^2} \\ 0 & 0 \end{bmatrix}.$$

Thus, $|F'(z)||z|$ calculated at $z = [1, 2]$ is equal to $[0.56, 0]$, while $F(z) = [0.25, 0]$ in $z = [1, 2]$. This shows that F is not strongly 1-subhomogeneus in $\mathbb{R}^n_{++}$.

Remark 3.3. Note that subhomogeneity and strong subhomogeneity coincide when the mapping has a positive valued subgradient. In fact, if $F \in \mathrm{subhom}_\mu(\mathbb{R}^n_+)$ and $\partial F(z) \subseteq \mathbb{R}^n_+$ for all $z \in \mathbb{R}^n_+$, then $|M\,z| = M\,z = |M|\,|z|$ for all $z \in \Omega$ and all $M \in \partial F(z)$. Thus, $F \in \text{s-subhom}_\mu(\mathbb{R}^n_+)$.

In order to better understand the connection between homogeneity and the two proposed notions of subhomogeneity, we provide below an analogous of Euler's theorem for subhomogeneous operators, directly connecting the notion of subhomogeneous operator with the usual notion of homogeneity $F(\alpha z) = \alpha^\mu F(z)$, see also (Lemmens & Nussbaum, 2012). The proof is deferred to Appendix A.

Proposition 3.4. *Let $F\colon \mathbb{R}^n \to \mathbb{R}^n$ be differentiable and Lipschitz. If $F \in \text{s-subhom}_\mu(\mathbb{R}^n_{++})$. Then, $F \in \mathrm{subhom}_\mu(\mathbb{R}^n_{++})$ and for all $\lambda \geq 1$ we have*

$$F(\lambda z) \leq \lambda^\mu F(z).$$

Assume moreover that $F'(z) > 0$, for all $z > 0$, i.e. the Jacobian $F'(z)$ is an entry-wise strictly positive matrix. Then, for $\mu > 0$, it holds

$$F \in \mathrm{subhom}_\mu(\mathbb{R}^n_{++}) = \text{s-subhom}_\mu(\mathbb{R}^n_{++})$$

if and only if for all $\lambda \geq 1$ we have

$$F(\lambda z) \leq \lambda^\mu F(z).$$

We provide below our main result showing uniqueness of fixed points for subhomogeneous operators, provided the homogeneity coefficient is small enough. Then, in Section 4, we will show that standard feed-forward linear layers of the form $\sigma(Wx)$ with a variety of broadly used activation functions σ are indeed subhomogeneous, sometimes up to a minor modification.

3.1. Main result: Existence and uniqueness of fixed points

In this section, we show that μ-subhomogeneus operators with small enough μ admit a unique fixed point. All proofs are moved to Appendix A.

The proof of the existence and uniqueness of the fixed point is based on the Banach fixed-point theorem and the following fundamental completeness result

Theorem 3.5 (See e.g. (Lemmens & Nussbaum, 2012))**.** *Let $R^n_{++} := \{z \in \mathbb{R}^n : z > 0,$ entrywise$\}$ denote the interior of the nonnegative orthant. Consider the Thomson distance $\delta(x, y) = \|\ln(x) - \ln(y)\|_\infty$. Then, the pair $(\mathbb{R}^n_{++}, \delta)$ is a complete metric space.*

Based on the theorem above, if we have an operator F that maps $\mathbb{R}^n_{++}$ to itself and that is contractive with respect to δ, then F must have a unique fixed point in $\mathbb{R}^n_{++}$. Our main theorem below shows that this is always the case when F is defined by a positive subhomogeneous operator. We start by proving that if F mapping $\mathbb{R}^n_{++}$ to $\mathbb{R}^m_{++}$ is subhomogeneous, then it is Lispchitz continuous with respect to the Thompson distance, with a Lispchitz constant that depends only on its subhomogeneity degree.

Theorem 3.6. *Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a Lipschitz operator. Assume that F is positive, i.e. $F(z) > 0$ for all $z > 0$, and that $F \in \mathrm{subhom}_\mu(\mathbb{R}^n_{++})$ for some $\mu > 0$. Then,*

$$\delta\left(F(x), F(y)\right) \leq \mu\, \delta(x, y),$$

for all $x, y \in \mathbb{R}^n_{++}$.

The following uniqueness result is now a direct consequence of Theorem 3.5 and Theorem 3.6.

Theorem 3.7. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be such that $F \in \mathrm{subhom}_\mu(\mathbb{R}^n_{++})$. If $0 < \mu < 1$, then there exists a unique fixed point $F(z^*) = z^* \in \mathbb{R}^n_{++}$. Moreover, the sequence $z_{k+1} = F(z_k)$ converges to z^* with linear convergence rate, namely*

$$\|z_k - z^*\|_\infty \leq C\mu^k\,,$$

for any $z_0 \in \mathbb{R}^n_{++}$.

In some occasions, for example when F is matrix-valued as in the context of graph neural networks, one may need to normalize rows or columns of the hidden embedding to e.g. simulate a random walk. See also Section 6.3.

In order to show the uniqueness of fixed points for neural networks with this type of normalization layer, we consider a general scaling function $\varphi : \mathbb{R}^n \to \mathbb{R}$ with the following properties:

- (1-homogeneous) $\varphi(\alpha z) = \alpha\varphi(z)$ for all coefficients $\alpha > 0$;
- (positive) $\varphi(z) > 0$ for each $z > 0$;
- (order-preserving) $\varphi(z) > \varphi(x)$ for each $z > x > 0$.

As $(\mathbb{R}^n_{++}, \delta)$ is a complete metric space, then also $(\mathbb{R}^n_{++}/\varphi, \delta)$ must be complete, where $R^n_{++}/\varphi := \{z \in \mathbb{R}^n_{++} : \varphi(z) = 1\}$. Our next result shows how Theorem 3.6 transfer to this setting

Theorem 3.8. *Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be as in Theorem 3.6. For a positive, 1-homogeneous, order-preserving functional $\varphi : \mathbb{R}^m \to \mathbb{R}$, define the φ-normalization layer map*

$$\mathrm{norm}_\varphi : \mathbb{R}^n_{++} \to \mathbb{R}^m_{++}/\varphi, \qquad \mathrm{norm}_\varphi(z) = z/\varphi(z)$$

and let $G(z) = \mathrm{norm}_\varphi(F(z))$. Then,

$$\delta\left(G(x), G(y)\right) \leq 2\mu\, \delta(x, y),$$

for all $x, y \in \mathbb{R}^n_{++}$. Moreover, if F is differentiable and its Jacobian matrix $F'(z)$ is entry-wise positive for all $z \in \mathbb{R}^n_{++}$, then

$$\delta\left(G(x), G(y)\right) \leq \mu\, \delta(x, y),$$

for all $x, y \in \mathbb{R}^n_{++}$,

The following equivalent of Theorem 3.7 is now a relatively direct consequence. We formulate it explicitly for normalization layers acting column-wise, but we underline that its proof (see Appendix A) can be easily adapted to different normalization patterns.

Theorem 3.9. *Let $G : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ be defined as*

$$G(x) = [\mathrm{norm}_{\varphi_1}(F_1(x)), \ldots, \mathrm{norm}_{\varphi_d}(F_d(x))],$$

where $F_i : \mathbb{R}^{n\times d} \to \mathbb{R}^n$ are μ-subhomogeneous and $\varphi_i : \mathbb{R}^n \to \mathbb{R}$ are 1-homogeneous, positive, order-preserving functions. If $0 < \mu < 1/2$, then there exists a unique fixed point $G(z^*) = z^*$. Moreover, if all the F_i are differentiable and their Jacobian matrices are entry-wise positive for all $z > 0$, then a unique fixed point exists provided $0 < \mu < 1$. In both cases, the sequence $z_{k+1} = G(z_k)$ converges to z with linear convergence rate, namely

$$\|z_k - z^*\|_\infty \leq C\mu^k\,,$$

for any $z_0 \in \mathbb{R}^n_{++}$.

4. Subhomogeneous deep equilibrium models

Consider now the weight-tied, input-injected neural network

$$z^{(k+1)} = \sigma_1(\,\sigma_2(Wz^{(k)}) + f_\theta(x)\,) \tag{3}$$

in which $x \in \mathbb{R}^d$ denotes the input, $z^{(k)} \in \mathbb{R}^n$ denotes the hidden unit at layer k, $\sigma_1, \sigma_2 : \mathbb{R} \to \mathbb{R}$ denote activation functions applied entrywise, $W \in \mathbb{R}^{n\times n}$ are the hidden unit weights, $f_\theta : \mathbb{R}^d \to \mathbb{R}^n$ is an input-injection embedding, e.g. defined by an MLP. We use here a fully connected layer for simplicity, but everything transfers unchanged to the convolutional case (i.e. if Wz is replaced by $W * z$, with W being the convolutional kernel). While in practice using both nonlinearities σ_1 and σ_2 may be redundant, we provide here a theoretical investigation of the model (3) in its generality and we will then study specific architectures where either $\sigma_1 = \mathrm{Id}$ or $\sigma_2 = \mathrm{Id}$.

In the following, we show that a unique fixed point for Equation (3) exists, when the number of layers k grows to infinity, provided the activation functions are subhomogeneous.

Consider the following DEQ architecture

$$z = \sigma_1(\sigma_2(Wz) + f_\theta(x)) \tag{4}$$

or the corresponding normalized version

$$z = \mathrm{norm}_\varphi\left(\,\sigma_1(\sigma_2(Wz) + f_\theta(x))\,\right) \tag{5}$$

where φ is any positive, 1-homogeneous, order-preserving normalizing function (or multiple functions if the normalization layer is applied locally as in Theorem 3.9). Note that φ can be, for example, any standard p-norm. This type of layer normalization step is also used to e.g. reduce the network sensitivity to small perturbations (Zhang et al., 2022; Farnia et al., 2018) and could accelerate training (Ba et al., 2016; Ioffe & Szegedy, 2015).

To study (4) and (5) using Theorem 3.7, we now notice that it is enough to study the subhomogeneity of the activation functions σ_1 and σ_2. In fact, we can think of $F(z) := \sigma_1(\sigma_2(Wz) + f_\theta(x))$ as the composition of an activation function σ_1 applied entry-wise, with a translation $T(u) = u + f_\theta(x)$, another activation function σ_2, and a linear map $L(z) = Wz$ (or $L(z) = W * z$ for convolutional layers),

$$F = \sigma_1 \circ T \circ \sigma_2 \circ L\,. \tag{6}$$

Linear mappings are particular examples of homogeneous operators while translations are subhomogeneous. In the next lemma, we observe that the subhomogeneity of F coincides with the subhomogeneity of σ_1 and σ_2, provided the input injection $f_\theta(x)$ is positive (entrywise). This requirement is not excessively restrictive, as it can be satisfied by using a positive nonlinear activation function, such as ReLU, $\mathrm{SoftPlus}$, or by shifting bounded activations such as $\tanh$. All proofs are moved to Appendix A.

Lemma 4.1. *Let Ω be a subset of $\mathbb{R}^n$, H be h-homogeneous, $P \in \mathrm{subhom}_\mu(H(\Omega))$, T_y denote the translation by y, $T_y(z) = z + y$, and let $\boldsymbol{Q} \in \text{s-subhom}_\lambda(T_y(P(H(\Omega))))$. Then, if the following composition rules hold:*

- $P \circ H \in \mathrm{subhom}_{h\mu}(\Omega)$
- *If $y \geq 0$, then $\boldsymbol{Q} \circ T_y \circ P \circ H \in \mathrm{subhom}_{h\mu\lambda}(\Omega)$*

Thus, studying the subhomogeneity (resp. strong subhomogeneity) of F in (6) boils down to the analysis of the subhomogeneity (resp. strong subhomogeneity) of σ_1 and σ_2. In particular, note that subhomogeneity is enough for σ_2 while strong homogeneity is required on σ_1 in order for the whole model to be subhomogeneous. However, we notice that in the typical case of activation functions acting entrywise this additional requirement is redundant. In fact, note that these subhomogeneity properties are inherited from the univariate function defining the activation layer when they act entrywise. Precisely, if $\sigma : \mathbb{R} \to \mathbb{R}$ is a Lipschitz function with $\sigma \in \mathrm{subhom}_\mu(\Omega)$, for $\Omega \subseteq \mathrm{dom}_+(\sigma)$, then we automatically have that the entrywise action of σ on $\mathbb{R}^n$ is $\mathrm{subhom}_\mu(\Omega^n)$, with $\Omega^n = \Omega \times \cdots \times \Omega$, as the elements of the Clarke's generalized Jacobian of σ in $z \in \mathbb{R}^n$ are simply diagonal matrices whose j-th diagonal element is an

element of the Clarke's generalized Jacobian of σ evaluated on the j-th component of z. With the following remark, we can notice that the definitions of subhomogeneous and strongly subhomogeneous coincide in the univariate case.

Remark 4.2. Let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be $\sigma \in \mathrm{subhom}_\mu(\Omega)$ for some $\mu > 0$ and some $\Omega \subseteq \mathbb{R}$. Notice that, $|\sigma'(z)\, z| = |\sigma'(z)||z|$ for each $z \in \Omega$, this implies that $\sigma \in \text{s-subhom}_\mu(\Omega)$.

In the next Section 5 we will show a variety of examples of commonly used activation functions σ that are subhomogeneous. For all such choices, the DEQ model (4) is well-defined as the existence of a unique equilibrium point z is guaranteed.

5. Subhomogeneous activation functions with corresponding subhomogeneity coefficient

We now propose several examples of subhomogeneous activation functions, they are well-known activation functions commonly used in deep learning. All the proofs are moved to Appendix A.

As a first example, we show that the $\mathrm{sigmoid}$ function is subhomogeneous, on the positive real axis, with $\mu = 1$.

Proposition 5.1. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be defined as*

$$\sigma(z) = \mathrm{sigmoid}(z) := \frac{e^z}{1 + e^z}.$$

Then $\sigma \in \mathrm{subhom}_1(\mathbb{R}_+)$.

Similarly to the $\mathrm{sigmoid}$, also the $\mathrm{SoftPlus}$ is subhomogeneous on $\mathbb{R}_+$ with $\mu = 1$.

Proposition 5.2. *Let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be defined as*

$$\sigma(z) = \mathrm{softplus}(z) := \frac{1}{\beta} \ln(1 + e^{\beta z}),$$

where $\beta > 0$. Then $\sigma \in \mathrm{subhom}_1(\mathbb{R}_+)$.

Also, the hyperbolic tangent is subhomogeneous in $\mathbb{R}_+$.

Proposition 5.3. *Let $\sigma\colon \mathbb{R} \to \mathbb{R}$ defined as*

$$\sigma(z) = \tanh(z) := \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Then, $\sigma \in \mathrm{subhom}_1(\mathbb{R}_+)$.

The examples considered above are activations that are subhomogeneous only on the nonnegative/positive orthant. We provide below an example of subhomogeneous activation function that is subhomogeneous globally. This is obtained as a positive shift of the hyperbolic tangent. In fact, it is not difficult to notice that if $\sigma(z) = \tanh(z) + 1 + \epsilon$ with $\epsilon > 0$, then $\mu(\epsilon) = \max_z |z|\sigma'(z)\sigma(z)^{-1}$ is a decreasing function of ϵ, i.e. $\mu(\epsilon_1) < \mu(\epsilon_2)$ for $\epsilon_1 > \epsilon_2 > 0$. Moreover, it holds $\mu(\epsilon) < 1$ for all

$$\epsilon > 0.199 > \max_z \left\{ |z|\mathrm{sech}^2(z) - \tanh(z) - 1 \right\}$$

and $\mu(\epsilon) < 1/2$ for all

$$\epsilon > 0.602 > \max_z \left\{ 2|z|\mathrm{sech}^2(z) - \tanh(z) - 1 \right\} .$$

These computations directly lead to

Proposition 5.4. *Let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be defined as*

$$\sigma(z) = \tanh(z) + \alpha\,.$$

If $\alpha > 1$, then $\sigma \in \mathrm{subhom}_\mu(\mathbb{R})$. In particular, if $\alpha > 1.2$ then $\mu < 1$ and if $\alpha > 1.602$ then $\mu < 1/2$.

Finally, similar to the hyperbolic tangent, we notice below that the non-differentiable piece-wise linear $\mathrm{hardtanh}$ activation function is 1-subhomogeneus on $\mathbb{R}$.

Proposition 5.5. *Let $\sigma : \mathbb{R}\colon \to \mathbb{R}$ be defined as*

$$\sigma(z) = \mathrm{hardtanh}(z) := \begin{cases} \alpha_1 & \text{if} \quad z < \alpha_1 \\ \alpha_2 & \text{if} \quad z > \alpha_2 \\ z & \text{otherwise} \end{cases}$$

with $0 < \alpha_1 < \alpha_2 < \infty$. Then, $\sigma \in \mathrm{subhom}_1(\mathbb{R})$.

Other two examples of 1-subhomogeneus operator over the whole real axis are the ReLU and the $\mathrm{LeakyReLU}$ with a slope $\alpha \leq 0$, both are positive 1-homogenous operator, thus they are also 1-subhomogeneus in $\mathbb{R}$.

All activation functions considered above are univariate functions acting entrywise. Thus they are both subhomogeneous and strongly subhomogeneous as observed in Remark 4.2. In the final example below we consider an example of a subhomogeneous function that is also strongly subhomogeneous but is not pointwise.

Proposition 5.6. *Let $\sigma : \mathbb{R}^n \to \mathbb{R}$ be defined as*

$$\sigma(z) = \mathrm{Approxmax}(z) = \ln \sum_i e^{z_i} \approx \max_i z_i$$

Then, $\sigma \in \mathrm{subhom}_1(\mathbb{R}^n_+)$ and $\sigma \in \text{s-subhom}_1(\mathbb{R}^n_+)$.

The power scaling trick

Theorem 3.7 ensures a unique fixed point for any subhomogeneous operator F exists, provided the subhomogeneity degree is small enough. As we have shown with the examples above, subhomogeneity is not a very stringent requirement, and many operators commonly used in deep learning are actually subhomogeneous without any modification. However, it is often the case that $\mu \geq 1$. To reduce the degree of $\tanh$, we applied a positive shift α. However, applying

Name	Sigmoid	SoftPlus	Tanh	Tanh + 1.2	Tanh + 1.603	HardTanh	LeakyReLU	Approxmax
μ	1	1	1	0.99	0.499	1	1	1
Ω	$\mathbb{R}^n_+$	$\mathbb{R}^n_+$	$\mathbb{R}^n_+$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n$	$\mathbb{R}^n_+$
Differentiable	✓	✓	✓	✓	✓			✓

Table 1. Examples of subhomogeneous activation functions

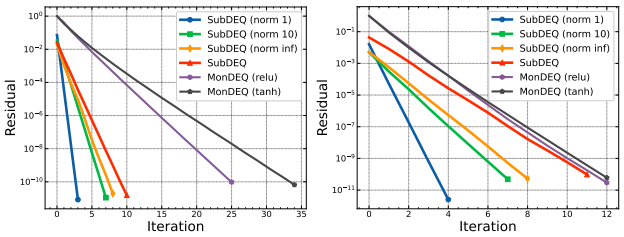

Figure 1. Iteration required by the fixed point method for SubDEQ vs Peaceman-Rachford method for MonDEQ. Left: linear layer; Right: convolutional layer.

Model	norm_φ	$\boldsymbol{\sigma} \in \text{subhom}_{\boldsymbol{\mu}}(\boldsymbol{\Omega})$ $\boldsymbol{\mu}$	$\boldsymbol{\Omega}$	Conditions on $\boldsymbol{W}$
$\boldsymbol{\sigma}(\boldsymbol{W}\,z)+y$	no	$\boldsymbol{\mu}<1$	$\boldsymbol{\Omega}=\mathbb{R}^n$	**None**
	yes	$\boldsymbol{\mu}<1/2$	$\boldsymbol{\Omega}=\mathbb{R}^n$	**None**
		$\boldsymbol{\mu}<1$	$\boldsymbol{\Omega}=\mathbb{R}^n_{++}$	$\boldsymbol{W}\geq 0$
$\boldsymbol{\sigma}(\boldsymbol{W}\,z+y)$	no	$\boldsymbol{\mu}<1$	$\boldsymbol{\Omega}=\mathbb{R}^n_{++}$	$\boldsymbol{W}\geq 0$
	yes	$\boldsymbol{\mu}<1$	$\boldsymbol{\Omega}=\mathbb{R}^n_{++}$	$\boldsymbol{W}\geq 0$

Table 2. Summary of requirements on weights and activations to guarantee existence and uniqueness of the DEQ fixed point, as well as the convergence of corresponding fixed point iteration. The setting requiring the fewest conditions is highlighted in gray color.

a scalar shift might not work all the time. When the subhomogeneity constant is not small enough, a simple "power scaling trick" can be implemented for any subhomogeneous F. In fact, it follows directly from Lemma 4.1 that if F is μ-subhomogeneous, then F^α defined as the entrywise power $F^\alpha(x) = F(x)^\alpha$, is $\alpha\mu$-subhomogeneous, since the map $x \mapsto x^\alpha$ is strongly α-subhomogeneous. Thus, to ensure uniqueness for (4) when σ_i are 1-subhomogeneous, we can mildly perturb σ_i into $\tilde{\sigma}_i(x) = \sigma_i(x)^{1-\varepsilon}$, for any $\varepsilon > 0$ arbitrary small. With this perturbed activation the DEQ in (4) is guaranteed to have a unique fixed point.

6. Experiments

All the models are implemented in PyTorch and the code is available at https://github.com/COMPiLELab/SubDEQ. We illustrate the performance of DEQ networks as in equations (4) and (5) on benchmark image datasets, as compared to alternative implicit neural networks, precisely Monotone operator-based DEQ (MonDEQ) (Winston & Kolter, 2020) and neural ordinary differential equations (nODE) (Chen et al., 2018), as well as standard explicit baseline architectures. As discussed in Section 4, we will use only one σ_i different from the identity function. To this end, we first consider the case $\sigma_1 = \text{Id}$, that is

$$z = \sigma(Wz) + f_\theta(x)$$

and

$$z = \text{norm}_\varphi(\,\sigma(Wz) + f_\theta(x)\,).$$

Thanks to Theorem 3.7, Theorem 3.9, and Lemma 4.1 we can choose a suitable activation function σ to guarantee the subhomogeneity of the architecture and thus the existence and uniqueness of the fixed point z, *without any assumption on* W, we summarize in 2 all the possible combination of well-posed implicit-layers. Here we experiment with $\sigma(z) = \tanh(z) + \alpha$, with α chosen accordingly to Proposition 5.4 to ensure a small enough subhomogeneity degree ($\mu < 1$ for the standard model and $\mu < 1/2$ for the normalized one) and thus uniqueness by Theorem 3.7. Precisely, we consider

$$z = \tanh(Wz) + f_\theta(x) + 1.2, \tag{7}$$

and

$$z = \text{norm}_{\|\cdot\|_p}(\,\tanh(Wz) + f_\theta(x) + 1.603\,), \tag{8}$$

where $1 \leq p \leq +\infty$ and $f_\theta(x)$ is a one-layer MLP with entry-wise positive final activation layer. As the architectures are now subhomogeneous and have a unique fixed point, we call (7) and (8) SubDEQ models.

6.1. Efficiency of SubDEQ

We now compare the convergence rate of the standard fixed–point method to find the equilibrium point of a SubDEQ, which is globally convergent due to Theorem 3.7, with the convergence rate of the Peaceman-Rachford method to find the equilibrium point of a MonDEQ (Winston & Kolter, 2020). We analyze the implicit–layer of the SubDEQ in (8) varying $p \in \{1, 10, +\infty\}$ and letting $f_\theta(x) = \text{ReLU}(Ux + b)$ be a one-layer MLP with ReLU activation function. We implement two different MonDEQs: the first one is defined as in (Winston & Kolter, 2020) via the following equation

$$z = \text{ReLU}(Wz + Ux + b)\,. \tag{9}$$

The second one is defined as

$$z = \tanh(Wz + f_\theta(x)) \,. \tag{10}$$

to replicate the architecture of the proposed SubDEQ.

In order to guarantee that the MonDEQ architectures have a unique fixed point, we restrict the weight matrix W in both the MonDEQ equations to satisfy the operator monotone parametrization

$$W = A^\top A + B - B^\top$$

with A, B parameter matrices, c.f. (Winston & Kolter, 2020).

For all models, as input, we choose $x \in \mathbb{R}^{128\times 400}$ sampled from a uniform distribution on the interval $[0, 1)$ using $\mathrm{torch.rand}$, the first dimension represents the batch–size and the second the dimension of the features space. The hidden embedding z has a width of 150.

In Figure 1 we plot the relative residual

$$\|z_{k+1} - z_k\|_F / \|z_{k+1}\|_F,$$

where $\|\cdot\|_F$ is the Frobenius norm and z_k are the iterates of the fixed point methods. From Figure 1(left) we can notice that SubDEQ systematically requires fewer steps to converge with respect to MonDEQ. We also compare with the case of convolutional layers, instead of dense DEQ layers. In this setting, the input $x \in \mathbb{R}^{128\times 1\times 28\times 28}$ is sampled from a uniform distribution on the interval $[0, 1)$ using $\mathrm{torch.rand}$, the first dimension represent the batch–size, the second the number of the image channels and the last two the size of the image. The hidden fixed point embedding z has 40 channels and each channel has a size of 28×28. The relative residual of the iterations is shown in Figure 1(right). Also in this case, we can notice that SubDEQ systematically requires fewer steps to converge to converge than MonDEQ.

6.2. SubDEQ on benchmark image datasets

We now show the capacity of SubDEQ in terms of classification tasks. We train them on different image benchmark datasets: CIFAR-10 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), and MNIST (LeCun & Cortes, 2010). We compare SubDEQ with MonDEQ as well as neural ode (nODE) architectures (Chen et al., 2018) and standard explicit network baselines. Overall, we consider the following models (for the feedforward implicit layer case):

1. SubDEQ (Normalized Tanh): $\mathrm{norm}_{\|\cdot\|_\infty}(\tanh(Wz) + f_\theta(x) + 1.603)$
2. SubDEQ (Tanh): $\tanh(Wz) + f_\theta(x) + 1.2$
3. MonDEQ (ReLU): $z = \mathrm{ReLU}(Wz + Ux + b)$
4. MonDEQ (Tanh): $z = \tanh(Wz + f_\theta(x))$
5. nODE (ReLU): $\dot{z}(t) = \mathrm{ReLU}(Wz(t))$, $z(0) = f_\theta(x)$

Model	Error %
MNIST (Dense)	
SubDEQ (Normalized Tanh)	2.088 ± 0.1405 %
SubDEQ (Tanh)	$\mathbf{1.92 \pm 0.102}$ **%**
nODE (ReLU)	2.356 ± 0.0689 %
nODE (Tanh)	3.296 ± 0.1082 %
MonDEQ (ReLU)	2.056 ± 0.0484 %
MonDEQ (Tanh)	2.736 ± 0.7491 %
Standard MLP (Tanh)	2.052 ± 0.1452 %
MNIST (Convolutional)	
SubDEQ (Normalized Tanh)	1.354 ± 0.98 %
SubDEQ (Tanh)	0.706 ± 0.011 %
nODE (ReLU)	1.184 ± 0.3845 %
nODE (Tanh)	0.826 ± 0.0432 %
MonDEQ (ReLU)	$\mathbf{0.654 \pm 0.0662}$ **%**
MonDEQ (Tanh)	1.096 ± 0.0589 %
Standard CNN (Tanh)	0.876 ± 0.0739 %
CIFAR-10	
SubDEQ (Normalized Tanh)	28.364 ± 0.377 %
SubDEQ (Tanh)	27.946 ± 1.7564 %
nODE (ReLU)	33.58 ± 1.2882 %
nODE (Tanh)	28.792 ± 1.3343 %
MonDEQ (ReLU)	$\mathbf{24.414 \pm 0.6521}$ **%**
MonDEQ (Tanh)	35.618 ± 1.1766 %
Standard CNN (Tanh)	27.157 ± 0.4154 %
SVHN	
SubDEQ (Normalized Tanh)	$\mathbf{9.3562 \pm 0.2122}$ **%**
SubDEQ (Tanh)	10.3987 ± 0.41296 %
nODE (ReLU)	33.8253 ± 11.3008 %
nODE (Tanh)	22.277 ± 2.8740 %
MonDEQ (ReLU)	11.0356 ± 0.319 %
MonDEQ (Tanh)	17.8849 ± 0.9747 %
Standard CNN (Tanh)	12.7243 ± 0.1024 %
TinyImageNet	
SubDEQ (Normalized Tanh)	$\mathbf{70.1791 \pm 0.3666}$ **%**
SubDEQ (Tanh)	74.6633 ± 0.1511 %
MonDEQ (ReLU)	85.49 ± 0.2406 %
Standard CNN (Tanh)	73.76 ± 1.4323 % %

Table 3. Mean $\pm$ std of the misclassification error on test set

6. nODE (Tanh): $\dot{z}(t) = \tanh(Wz(t))$, $z(0) = f_\theta(x)$.

We also consider a convolutional variant of these implicit layers, the only difference with the above dense layers is in the weights matrices, which we substitute with the standard convolutional kernels. For the standard explicit network baseline, we replace the implicit layer with a standard one using the same hyperparameter and $\tanh$ as the activation function, in both, the dense and the convolutional case. For the normalized SubDEQ as in (8), we decide to normalize each element of the batch for the feedforward, while with the convolutions we normalize along each row. For the

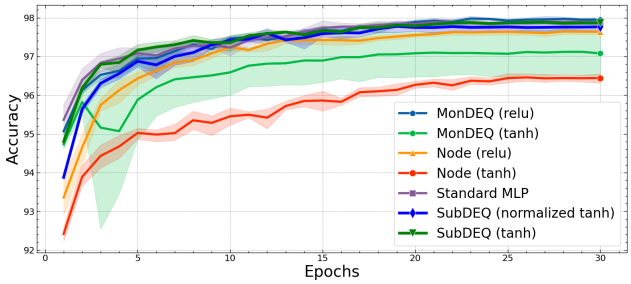

Figure 2. Validation accuracy of the dense architectures during training on MNIST.

nODE, we integrate the ODE over the interval $[0, 1]$, as the output of the implicit layer we took the solution at time $t = 1$. For all the models, we apply 1d batch normalization for the feedforward layers and 2d batch normalization for the convolutional layers. Moreover, as a last step, we feed the embedding into a softmax classifier, but with the convolutional architectures, before it, we apply an average pooling and then we flatten the tensor. We describe all the details about the hyperparameters in Table 8 in Appendix C.

For every experiment, with the SubDEQ architectures, we use the fixed point method speed up with Anderson acceleration to calculate the fixed point and as the stopping criterion we use the relative residual

$$\|z_{k+1} - z_k\|_F / \|z_{k+1}\|_F,$$

and we stop the method when the residual reaches the values of 10^{-3}. We decided to train the dense architecture only on MNIST, instead, we trained the convolutional models on all three data sets. Regarding the data, we use the hold-out approach, dividing the dataset into train validation and test sets. Appendix C describes the proportion of the splittings. All training data is normalized to mean $\mu = 0$, standard deviation $\sigma = 1$, and the validation and the test are rescaled using the mean and standard deviation of the training data. On the same data split we run the experiments 5 times and in Table 3 we show the mean $\pm$ the standard deviation of the misclassification error. As we can notice from Table 3, the SubDEQ (Normalized Tanh), depending on the dataset, achieves performance that exceeds the performance of the other model or reaches similar performance to the best model. We highlight that adding the vector 1.603 to the activation function to ensure the uniqueness and convergence, does not negatively affect the performance, instead ensures higher stability and leads the model to a higher accuracy.

6.3. Deep Equilibrium Graph Neural Network: nonlinear graph propagation

We conclude with an experiment on a graph neural network architecture. Graph neural networks are typically relatively shallow due to oversmoothing and oversquashing phenomena (Nguyen et al., 2023; Giraldo et al., 2023). Thus, a fixed point implicit graph neural architecture may be ineffective. However, it is shown in (Gasteiger et al., 2018) that the PageRank random walk on the graph may be used to propagate the input injected by a simple MLP and the limit point of the propagation leads to excellent, sometimes state-of-the-art, results. This approach, named there APPNP, is effective also because the PageRank diffusion process converges linearly to a unique PageRank vector. As the model uses the PageRank fixed point as the final latent embedding, it can be interpreted as a simple DEQ with linear activations. This is possibly a limitation of APPNP as nonlinear activations may allow for a better modeling power. Using Theorem 3.7 one can propose variations of this approach implementing nonlinear diffusion processes based on subhomogeneous activation functions and still maintain the same fundamental guarantees of uniqueness and linear convergence. We obtain this way a SubDEQ variation of the APPNP graph neural network model, which we discuss next.

Dataset (% labeled)	**Accuracy**
Cora citation (5.2 %)	
APPNP	$\mathbf{80.2027 \pm 1.9557\%}$
APPNP (Normalized Tanh)	73.3905 ± 2.3818 %
APPNP (Tanh)	72.2369 ± 2.3013 %
Cora author (5.2%)	
APPNP	70.834 ± 2.1591 %
APPNP (Normalized Tanh)	$\mathbf{73.3437 \pm 1.8252}$ **%**
APPNP (Tanh)	72.5175 ± 2.4537 %
CiteSeer (4.2 %)	
APPNP	$\mathbf{62.8625 \pm 1.4477}$ **%**
APPNP (Normalized Tanh)	62.66078 ± 1.8676 %
APPNP (Tanh)	62.1564 ± 1.479 %
DBLP (4.0 %)	
APPNP	88.8095 ± 0.2866
APPNP (Normalized Tanh)	$\mathbf{89.4007 \pm 0.3619}$ **%**
APPNP (Tanh)	86.87046 ± 0.3851 %
PubMed (0.8%)	
APPNP	77.45168 ± 1.4433 %
APPNP (Normalized Tanh)	$\mathbf{78.5827 \pm 0.9741}$ **%**
APPNP (Tanh)	77.103 ± 1.251 %

Table 4. Mean $\pm$ standard deviation accuracy on test set

First, we review the standard APPNP architecture (Gasteiger et al., 2018). Consider an undirected graph $G = (V, E)$ on $n = |V|$ nodes, let d_i be the degree of node i, and let A be the graph normalized adjacency matrix, defined as $A_{ij} = d_i$ if $ij \in E$ and $A_{ij} = 0$ otherwise. APPNP defines the node embedding Z via the fixed point equation:

$$\begin{cases} \tilde{Z} = (1-\alpha)\tilde{A}\tilde{Z} + \alpha f_\theta(X) \\ Z = \mathrm{softmax}(\tilde{Z}) \end{cases} \tag{11}$$

where $\tilde{A} = A + I$ denotes the shifted adjacency matrix and $X \in \mathbb{R}^{n \times f}$ is the input node feature matrix.

We consider here the following nonlinear

variations of (11)

$$\begin{cases} \tilde{Z} = \tanh\left((1-\alpha)\tilde{A}\tilde{Z}\right) + \alpha f_\theta(X) + 1.2 \\ Z = \mathrm{softmax}(\tilde{Z}) \end{cases}$$

and

$$\begin{cases} \tilde{Z} = \mathrm{norm}_{\|\cdot\|_\infty}\left(\tanh\left((1-\alpha)\tilde{A}\tilde{Z}\right) + \alpha f_\theta(X) + 1.2\right) \\ Z = \mathrm{softmax}(\tilde{Z}) \end{cases}$$

where the final l^∞ normalization layer is implemented columnwise in the model above, as in Theorem 3.9.

If we add to $\tanh$ a translation vector with all entries equal to 1.2, we obtain a subhomogeneous operator with degree $\mu = 0.99$, due to the Proposition 5.4 and Lemma 4.1. Notice also that the Jacobian with respect Z of this transformation is entrywise positive respect all $Z > 0$ and is differentiable. Thus, both the nonlinear fixed point equations above have a unique fixed point due to Theorem 3.9. We test APPNP and its nonlinear variations on different graph datasets: Cora citation, Cora author, CiteSeer, DBLP, and PubMed, always in a node classification semi-supervised learning setting. We divide the dataset into training, validation, and test sets. The percentages of observation used for the training set are shown in Table 4, and the remaining observations are equally split between validation and test sets. For both methods, we use similar hyperparameters, as $f_\theta(\cdot)$ we use 2-layers MLP of width 64 for each layer; we regularize the architectures with dropout $p = 0.5$, we use Adam as optimizer with constant learning rate equal to 0.01 and a weight decay equal to 0.005. We set $\alpha = 0.1$ and $K = 10$. We repeat the splitting and the training 5 times and we report the average $\pm$ std results in Table 4.

7. Conclusions

We have presented a new analysis of the existence and uniqueness of fixed points for DEQ models, as well as convergence guarantees for the corresponding fixed point iterations. Unlike previous approaches that require constraints on the weight matrices to guarantee uniqueness, our theoretical framework allows us to use general weight matrices, provided the activation functions of the network are subhomogeneous. We observe that many well-known activation functions are indeed subhomogeneous possibly up to some minor modification, showing the vast applicability of our framework. Thus, we provide several examples of new subhomogeneous deep equilibrium architectures designed for image classification and nonlinear graph propagation.

Impact statement

This paper present work whose goal is to advance the field of implicit-depth deep learning architectures from a mathematical point of view. There are many pontential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *stat*, 1050:21, 2016.

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. *Advances in neural information processing systems*, 32:688–699, 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.

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. doi: 10.1109/TIP.2023.3255104.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. *Advances in neural information processing systems*, 31:6571–6583, 2018.

Chen, T., Lasserre, J. B., Magron, V., and Pauwels, E. Semi-algebraic representation of monotone deep equilibrium models and applications to certification. *Advances in Neural Information Processing Systems*, 34:27146–27159, 2021.

Clarke, F. H. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990. doi: 10.1137/1.9781611971309. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611971309.

Dabre, R. and Fujita, A. Recurrent stacking of layers for compact neural machine translation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6292–6299, 2019.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, L. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 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.

Farnia, F., Zhang, J., and Tse, D. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2018.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2018.

Gautier, A., Tudisco, F., and Hein, M. The perron–frobenius theorem for multihomogeneous mappings. *SIAM Journal on Matrix Analysis and Applications*, 40(3):1179–1205, 2019.

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

Giraldo, J. H., Skianis, K., Bouwmans, T., and Malliaros, F. D. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23. ACM, October 2023. doi: 10.1145/3583780.3614997. URL http://dx.doi.org/10.1145/3583780.3614997.

Gu, F., Chang, H., Zhu, W., Sojoudi, S., and El Ghaoui, L. Implicit graph neural networks. *Advances in Neural Information Processing Systems*, 33:11984–11995, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Jafarpour, S., Davydov, A., Proskurnikov, A., and Bullo, F. Robust implicit networks via non-euclidean contractions. *Advances in Neural Information Processing Systems*, 34: 9857–9868, 2021.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lemmens, B. and Nussbaum, R. *Nonlinear Perron–Frobenius Theory*. Cambridge Tracts in Mathematics. Cambridge University Press, 2012. doi: 10.1017/CBO9781139026079.

Micaelli, P., Vahdat, A., Yin, H., Kautz, J., and Molchanov, P. Recurrence without recurrence: Stable video landmark detection with deep equilibrium models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22814–22825, 2023.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Nguyen, K., Hieu, N. M., Nguyen, V. D., Ho, N., Osher, S., and Nguyen, T. M. Revisiting over-smoothing and over-squashing using ollivier-ricci curvature. In *International Conference on Machine Learning*, pp. 25956–25979. PMLR, 2023.

Piotrowski, T., Cavalcante, R. L., and Gabor, M. Fixed points of nonnegative neural networks. *arXiv e-prints*, pp. arXiv–2106, 2021.

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

Rusch, T. K. and Mishra, S. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2021.

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

Wei, C. and Kolter, J. Z. Certified robustness for deep equilibrium models via interval bound propagation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=y1PXylgrXZ.

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

Yang, Z., Pang, T., and Liu, Y. A closer look at the adversarial robustness of deep equilibrium models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 10448–10461. Curran Associates, Inc., 2022.

Zhang, B., Jiang, D., He, D., and Wang, L. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. *Advances in neural information processing systems*, 35:19398–19413, 2022.

Zhao, Y., Zheng, S., and Yuan, X. Deep equilibrium models for video snapshot compressive imaging. *arXiv preprint arXiv:2201.06931*, 2022.

A. Proofs

Proof. (Proposition 3.4)
We now prove the first part. Let $g\colon [1,+\infty) \to \mathbb{R}^n$, defined as

$$g(\lambda) = [g_1(\lambda), \dots, g_n(\lambda)] = F(\lambda z) - \lambda^\mu F(z).$$

Clearly, $g(1) = 0$ and g is a differentiable function since $F(z) > 0$ for each $z > 0$ and F is differentiable.

$$\begin{aligned} g'(\lambda) &= F'(\lambda z)z - \mu\lambda^{\mu-1}F(z) \\ \lambda g'(\lambda) &= F'(\lambda z)\lambda z - \mu\lambda^\mu F(z) \le |F'(\lambda z)|\lambda z - \mu\lambda^\mu F(z). \end{aligned}$$

Using the definition of subhomogeneous operator we get

$$\lambda g'(\lambda) \le \mu(F(\lambda z) - \lambda^\mu F(z)) = \mu g(\lambda).$$

Then $g_j'(\lambda) \le \frac{\mu}{\lambda} g_j(\lambda)$ for each $j = 1, \dots, n$, thanks to the Grönwall's inequality,

$$g_j'(\lambda) \le g_j(1) \exp\left(\int_1^\lambda \frac{\mu}{t} dt\right) = g_j(1)\lambda = 0.$$

Therefore, $g_j'(\lambda) \le 0$, this shows that g_j is a decreasing function and $g_j(\lambda) \le 0$, thus $g(\lambda) \le 0$ entry-wise.
We now prove the second part of the proposition, the necessary condition is implied by the first part so we will prove only the sufficient condition.
Let $g\colon [0,+\infty) \to \mathbb{R}^n$ be defined as $g(\lambda) = F(\lambda z) - \lambda^\mu F(z)$, by hypothesis, for each $\lambda \ge 1$, $g(\lambda) \le 0$, also note that $g(1) = 0$. This shows that $g'(\lambda) \le 0$ for each $\lambda \ge 1$.

$$g'(\lambda) = F'(\lambda z)z - \mu\lambda^{\mu-1}F(z),$$

we get

$$g'(1) = F'(z)z - \mu F(z),$$

which implies $F'(z)z \le \mu F(z)$. □

Proof. (Theorem 3.6)
Let $D(z)$ be Clark's generalized Jacobian of the map $z \to \ln(G(e^z))$. For the Mean Theorem (Clarke, 1990) we have

$$\ln\left(F\left(e^{\ln(x)}\right)\right) - \ln\left(F\left(e^{\ln(y)}\right)\right) \in co\left(D(\Omega(x,y))\left(\ln(x) - \ln(y)\right)\right),$$

where $\Omega(x,y)\colon = \{z \in \mathbb{R}^n \mid z = t\ln(x) + (1-t)\ln(y),\ t \in [0,1]\}$ and $co\left(D(\Omega(x,y))\left(\ln(x) - \ln(y)\right)\right)$ denotes the convex hull of all points of the form $Z(ln(x) - ln(y))$ where $Z \in D(u)$ for some u in $\Omega(x,y)$.
For the Caratheodory Theorem, we get

$$\ln\left(F\left(e^{\ln(x)}\right)\right) - \ln\left(F\left(e^{\ln(y)}\right)\right) = \sum_{l=0}^{n^2} \beta_j \xi_l (\ln(x) - \ln(y)),$$

where $\xi_l \in D(u)$ for a u in $\Omega(x,y)$, $\beta_l \ge 0$ and $\sum_l \beta_l = 1$. Let $v = \ln(x) - \ln(y)$, using the Chain Rule (Clarke, 1990), we obtain

$$D(u)v \subseteq \left(\mathrm{Diag}(F(e^u))^{-1}\partial F(e^u)\,\mathrm{Diag}(e^u)\right)v,$$

$\partial F(u)$ detonates the Clark's Generalized Jacobian of F with respect u. Therefore, we can write the element of $D(u)v$ as

$$\mathrm{Diag}(F(e^u))^{-1} Q\,\mathrm{Diag}(e^u)v,$$

where $Q \in \partial F(u)$. At this point, we can estimate $\delta(G(x), G(y))$ as follows,

$$\begin{aligned} \delta(G(x), G(y)) &= \|\ln(G(x)) - \ln(G(y))\|_\infty = \|\sum_{j=0}^{n^2} \beta_j \xi_j (\ln(x) - \ln(y))\|_\infty \le \\ &\le \sum_{l=0}^{n^2} \beta_l \|\xi_l\|_\infty \|(\ln(x) - \ln(y))\|_\infty \le \max_{l=0,\dots,n^2} \|\xi_l\|_\infty \|(\ln(x) - \ln(y))\|_\infty. \end{aligned} \tag{12}$$

Moreover, using the definition of subhomogeneous operator we obtain

$$\|\xi_l\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |\xi_l|_{ij} = \max_{i=1,\dots,n} \sum_{j=1}^{n} |\operatorname{Diag}(F(e^u))^{-1} Q \operatorname{Diag}(e^u)|_{ij} =$$
$$= \max_{i=1,\dots,n} \sum_{j=1}^{n} \left| \frac{1}{F(e^u)_i} Q_{ij}\, e^u_j \right| = \max_{i=1,\dots,n} \frac{1}{F(e^u)_i} \sum_{j=1}^{n} \left| Q_{ij}\, e^u_j \right| \leq \max_{i=1,\dots,n} \frac{1}{F(e^u)_i} F(e^u)_i\, \mu = \mu.$$

□

Proof. (Theorem 3.8)
For the Euler's Homogeneous Function Theorem $\varphi(z) = w_z^\top z$, thus $G(z) = \frac{F(z)}{w_z^\top F(z)}$. Let $D(z)$ be Clark's generalized Jacobian of the map $z \to \ln(G(e^z))$. For the Mean Theorem (Clarke, 1990) we have

$$\ln(G(e^{\ln(x)})) - \ln(G(e^{\ln(y)})) \in co\,(D(\Omega(x,y))\,(\ln(x) - \ln(y)),$$

where $\Omega(x,y) \colon = \{z \in \mathbb{R}^n \mid z = t\ln(x) + (1-t)\ln(y),\ t \in [0,1]\}$ and $co\,(D(\Omega(x,y))\,(\ln(x) - \ln(y))$ denote the convex hull of all points of the form $Z(ln(x) - ln(y))$ where $Z \in D(u)$ for some u in $\Omega(x,y)$.
For the Caratheodory Theorem, we get

$$\ln(G(e^{\ln(x)})) - \ln(G(e^{\ln(y)})) = \sum_{j=0}^{nm} \beta_j \xi_j (\ln(x) - \ln(y)).$$

where $\xi_j \in D(u)$ for a u in $\Omega(x,y)$, $\beta_j \geq 0$ and $\sum_i \beta_i = 1$. For simplicity we will pose $v = \ln(x) - \ln(y)$. Using the Chain Rule (Clarke, 1990) we obtain

$$D(u)v \subseteq \left(\operatorname{Diag}(G(e^u))^{-1} \partial G(e^u) \operatorname{Diag}(e^u)\right) v,$$

$\partial G(u)$ detonates the Clark's Generalized Jacobian of G with respect u. Since $G(x) = \frac{F(x)}{w_u^\top F(x)}$ and the generalized Jacobian of w_u is zero, if we apply the chain rule (Clarke, 1990) several times we obtain

$$\partial G(e^u) \operatorname{Diag}(e^u) v \subseteq \left(\frac{\partial F(e^u)}{w_u^\top F(e^u)} - \frac{F(e^u) w_u^\top \partial F(e^u)}{(w_u^\top F(e^u))^2} \right) \operatorname{Diag}(e^u) v,$$

the right-hand side above denote the set of points $Q \operatorname{Diag}(e^u)\, v$ where $Q = \frac{H}{w_u^\top F(e^u)} - \frac{F(e^u) w_u^\top K}{(w_u^\top F(e^u))^2}$ and $H, K, \in \partial F(e^u)$. Therefore we can write the element of $D(u)v$ as

$$\operatorname{Diag}(G(e^u))^{-1} Q \operatorname{Diag}(e^u) v.$$

At this point, we will estimate $\delta(G(x), G(y))$ using the calculations that we have done so far.

$$\delta(G(x), G(y)) = \|\ln(G(x)) - \ln(G(y))\|_\infty = \|\sum_{j=0}^{mn} \beta_j \xi_j (\ln(x) - \ln(y))\|_\infty \leq$$
$$\leq \sum_{j=0}^{mn} \beta_j \|\xi_j\|_\infty \|(\ln(x) - \ln(y))\|_\infty \leq \max_{j=0,\dots,mn} \|\xi_j\|_\infty \|(\ln(x) - \ln(y))\|_\infty. \tag{13}$$

Now we estimate the infinity norm of a matrix of the form $\operatorname{Diag}(G(e^u))^{-1} Q \operatorname{Diag}(e^u)$. Let us begin by focusing on the first two matrices of multiplication, thus

$$\operatorname{Diag}(G(e^u))^{-1} Q = \operatorname{Diag}(F(e^u))^{-1} w_u^\top F(e^u) Q =$$
$$= \operatorname{Diag}(F(e^u))^{-1} H - \frac{\mathbf{1} w_u^\top K}{w_u^\top F(e^u)},$$

consequently, the entries are

$$\left|\mathrm{Diag}(G(e^u))^{-1}Q\right|_{ij} = \left|\mathrm{Diag}(F(e^u))^{-1}H - \frac{\mathbf{1}\, w_u^\top K}{w_u^\top P(e^u)}\right|_{ij} = \left|\frac{H_{ij}}{F(e^u)_i} - \sum_{l=1}^{m} \frac{w_{u,l}K_{lj}}{\sum_r w_{u,r}F(e^u)_r}\right| =$$
$$= \left|\frac{H_{ij}}{F(e^u)_i} - \sum_{l=1}^{m} \frac{w_{u,l}F(e^u)_l}{\sum_r w_{u,r}F(e^u)_r}\frac{K_{lj}}{F(e^u)_l}\right| = \left|\frac{H_{ij}}{F(e^u)_i} - \sum_{l=1}^{m} \gamma_l \frac{K_{lj}}{F(e^u)_l}\right|,$$

where $\gamma_i = \frac{w_{u,i}F(e^u)_i}{\sum_r w_{u,r}F(e^u)_r}$. Notice that γ_i are positive and $\sum_i \gamma_i = 1$.

$$\begin{aligned}
\|\,\mathrm{Diag}(G(e^u))^{-1}Q\,\mathrm{Diag}(e^u)\|_\infty &= \max_{i=1,\dots,m} \sum_{j=1}^{n} |\,\mathrm{Diag}(G(e^u))^{-1}Q\,\mathrm{Diag}(e^u)|_{ij} = \\
&= \max_{i=1,\dots,m} \sum_{j=1}^{n} \left|\frac{H_{ij}e^u_j}{F(e^u)_i} - \sum_{l=1}^{m}\gamma_l \frac{K_{lj}e^u_j}{F(e^u)_i}\right| \le \\
&\le \max_{i=1,\dots,m}\left|\frac{(He^u)_i}{F(e^u)_i}\right| + \max_{i=1,\dots,m}\left|\frac{(Ke^u)_i}{F(e^u)_i}\right| \le \mu + \mu = 2\mu. \qquad (14)
\end{aligned}$$

Finally we obtain $\max_{j=0,\dots,mn} \|\xi_j\|_\infty \le 2\mu$, that conclude the proof of the first part. We now assume that F is differentiable and its Jacobian is entrywise positive for each $z > 0$. Until Equation (13) the proof is the same as the first part. Therefore, we can start estimating $\mathrm{Diag}(G(e^u))^{-1}Q\,\mathrm{Diag}(e^u)$ as following

$$\left|\mathrm{Diag}(G(e^u))^{-1}Q\right|_{ij} = \left|\mathrm{Diag}(F(e^u))^{-1}J_F(e^u) - \frac{\mathbf{1}w_u^\top J_F(e^u)}{w_u^\top F(e^u)}\right|_{ij} = \left|\frac{J_F(e^u)_{ij}}{F(e^u)_i} - \sum_{l=1}^{m}\frac{w_{u,l}J_F(e^u)_{lj}}{\sum_r w_{u,r}F(e^u)_r}\right| =$$
$$= \left|\frac{J_F(e^u)_{ij}}{F(e^u)_i} - \sum_{l=1}^{m}\frac{w_{u,l}F(e^u)_l}{\sum_r w_{u,r}F(e^u)_r}\frac{J_F(e^u)_{lj}}{F(e^u)_l}\right| =: \left|C_{ij} - \sum_{l=1}^{m}\gamma_l C_{lj}\right|$$

where $\gamma_i = \frac{w_{u,i}F(e^u)_i}{\sum_r w_{u,r}F(e^u)_r}$ and $C_{ij} = \frac{J_F(e^u)_{ij}}{F(e^u)_i}$. Notice that γ_i and C_{ij} are positive and $\sum_i \gamma_i = 1$.

$$|C_{ij} - \sum_{l=1}^{m}\gamma_l C_{lj}| \le \max_{l=1,\dots,m} C_{lj} =: C_{\bar{l}j}.$$

Thus

$$\begin{aligned}
\|\,\mathrm{Diag}(G(e^u))^{-1}Q\,\mathrm{Diag}(e^u)\|_\infty &= \max_{i=1,\dots,m}\sum_{j=1}^{n}|\,\mathrm{Diag}(G(e^u))^{-1}Q\,\mathrm{Diag}(e^u)|_{ij} = \\
&= \max_{i=1,\dots,m}\sum_{j=1}^{n}|\,\mathrm{Diag}(G(e^u))^{-1}Q|_{ij}e^u_j \le \\
&\le \sum_{j=1}^{n} C_{\bar{l}j}e^u_j = \sum_{j=1}^{n}\frac{J_F(e^u)_{\bar{l}j}}{F(e^u)_{\bar{l}}}e^u_j \le \mu.
\end{aligned}$$

That conclude the proof. □

Proof. (Theorem 3.9)
Let

$$\delta(G(x), G(y)) = \max_{\substack{i=1,\dots,n\\ j=1,\dots,d}} |\ln(x_{ij}) - \ln(y_{ij})|$$

be the Thompson distance on $\mathbb{R}^{n\times d}$. Let $G_j : \mathbb{R}^{n\times d} \to \mathbb{R}^n$ be the j-th column mapping of G.

$$\delta(G(x), G(y)) = \max_{\substack{i=1,\dots,n\\ j=1,\dots,d}} |\ln(x_{ij}) - \ln(y_{ij})| = \max_{j=1,\dots,d}\left(\max_{i=1,\dots,n}|\ln(x_{ij}) - \ln(y_{ij})|\right) = \max_{j=1,\dots,d}\delta(G_j(x), G_j(y))\,.$$

Applying Theorem 3.8 with input space $\mathbb{R}^{n\times d}$ and output space $\mathbb{R}^n$, we obtain

$$\delta\left(G_j(x), G_j(y)\right) \leq 2\mu\delta(x,y)$$

for all $j = 1, \ldots, d$. Thus

$$\max_{j=1,\ldots,d} \delta\left(G_j(x), G_j(y)\right) \leq 2\mu\delta(x,y).$$

This concludes the proof of the first part. Applying the same argument one can prove the case of positive Jacobian yielding the same upper bound without 2. □

Proof. (Lemma 4.1)
Now we start proving the first part, hence that $P\circ H$ is $h\mu$-subhomogeneous. Using the chain rule we obtain $\partial(P\circ H)(z)z = \partial P(H(z))J_H(z)z$, where $\partial(P\circ H)(z)$ and $\partial P(H(z))$ are the Clarke's generalized Jacobians of P and $P\circ H$, respectively evaluated in z and $P(z)$, and $J_H(z)$ is the Jacobian of H in z.
Therefore we can write an element of $\partial(P\circ H)(z)z$ as $MJ_H(z)z$, with $M \in \partial P(z)$. Moreover, applying Euler's Homogeneous Function Theorem and the definition of subhomogeneity we get

$$|MJ_H(z)z| = h|MH(z)| \leq h\mu P(H(z)) = h\mu(P\circ H)(z).$$

Thus, $P\circ H \in \mathrm{subhom}_{h\mu}$.
We now prove the second part of the Lemma, hence $Q\circ T_y\circ P\circ H$ is $h\mu\lambda$-subhomogeneous. Let $F = Q\circ T_y\circ P\circ H$ and $\boldsymbol{P} = P\circ H$, for the first point is $h\mu$-subhomogeneous. let $\partial F(z)z \subseteq co\{\partial Q(\boldsymbol{P}(z)+y)\partial\boldsymbol{P}(z)\}z$. Thus, an element $M \in \partial F(z)$ can be written as

$$M\,z = \sum_{k=0}^{n} \beta_k M_k^Q M_k^{\boldsymbol{P}}\, z,$$

where $\beta_k \geq 0$ and $\sum_k \beta_k = 1$, $M_k^Q \in \partial Q(\boldsymbol{P}(z)+y)$ and $M_k^P \in \partial\boldsymbol{P}(z)$. We get

$$|M\,z| = |\sum_{k=0}^{n} \beta_k M_k^Q M_k^{\boldsymbol{P}}\, z| \leq \lambda \sum_{k=0}^{n} \beta_k |M_k^Q| \boldsymbol{P}(z) \leq \lambda \sum_{k=0}^{n} \beta_k |M_k^Q| (\boldsymbol{P}(z)+y) \leq \lambda\mu Q(\boldsymbol{P}(z)+y) = \lambda\mu F(z).$$

Therefore, $Q\circ T_y\circ P\circ H \in \mathrm{subhom}_{h\mu\lambda}$. □

Proof. (Proposition 5.1)
If $z \geq 0$, $\sigma(z) > 0$, then $\mathbb{R}_+ \subseteq \mathrm{dom}_+(\sigma)$. Also note that

$$\sigma'(z) = \frac{e^z(1+e^z) - e^z e^z}{(1+e^z)^2} = \sigma(z)(1-\sigma(z)).$$

Since $\sigma(z) = \frac{e^z}{1+e^z}$, the inequality that we want to prove is equivalent to $z \leq 1+e^z$. In 0 the inequality is verified, moreover, if we calculate the derivative of both sides we obtain

$$1 \leq e^z,$$

$e^0 = 1$ and e^z is a monotone increasing function, thus the inequality is verified. Therefore $\sigma \in \mathrm{subhom}_1(\mathbb{R}_+)$. □

Proof. (Proposition 5.2)
If $z \geq 0$, $\sigma(z) > 0$, thus $\mathbb{R}_+ \subseteq \mathrm{dom}_+(\sigma)$.

$$\sigma'(z) = \frac{e^{\beta z}}{1+e^{\beta z}},$$

$0 < \sigma'(z) < 1$, thus σ is a monotonic increasing function. Therefore what we want to show coincides with

$$\sigma'(z)x \leq \sigma(z).$$

Moreover, $\sigma'(z)z \leq x$, since $0 < \sigma'(z) < 1$. Clearly is enough to proof $\beta z \leq ln(1+e^{\beta z})$, this is equivalent to $e^{\beta z} \leq 1+e^{\beta z}$ that holds for all $z \in \mathbb{R}_+$. □

Proof. (Proposition 5.3)
Note that, if $z \geq 0$, $\sigma(z) > 0$, then $\mathbb{R}_+ \subseteq \mathrm{dom}_+(\sigma)$.

$$\sigma'(z) = \frac{4}{(e^z + e^{-z})^2}. \tag{15}$$

Since, if $z \geq 0$ $\sigma'(z) > 0$, what we want to prove is equivalent to $\sigma'(z)\, z \leq \sigma(z)$. If we plug in the last equation the exponential formulation of the hyperbolic tangent and 15 we get

$$\frac{4z}{e^z + e^{-z}} \leq e^z - e^{-z}.$$

Using the exponential formulation of the hyperbolic sine we obtain

$$2z \leq \sinh(2z).$$

The Taylor expansion of the $\sinh$ is

$$\sinh(2z) = 2z + \frac{(2z)^3}{3!} + \frac{(2z)^5}{5!} + \frac{(2z)^7}{7!} + \cdots = \sum_{k=0}^{\infty} \frac{(2z)^{2k+1}}{(2k+1)!}.$$

Since the first term of the series is $2z$ and $z \in \mathbb{R}_+$, the inequality is verified. Thus $\tanh \in \mathrm{subhom}_1(\mathbb{R}_+)$. □

Proof. (Proposition 5.5)
First, note that, for each $z \in \mathbb{R}$, $\sigma(z) > 0$, thus $\mathrm{dom}_+(\sigma) = \mathbb{R}$. The Clarke's generalized Jacobian of H respect to z is:

$$\partial\sigma(z) = \begin{cases} 0 \quad \text{if} \quad z < \alpha_1 \quad \text{or} \quad z > \alpha_2, \\ [0,1] \quad \text{if} \quad z = \alpha_1 \quad \text{or} \quad z = \alpha_2 \\ 1 \quad \text{otherwise} \end{cases}$$

Then we get

$$\partial\sigma(z)z = \begin{cases} 0 \quad \text{if} \quad z < \alpha_1 \quad \text{or} \quad z > \alpha_2, \\ [0,\alpha_1] \quad \text{if} \quad z = \alpha_1 \\ [0,\alpha_2] \quad \text{if} \quad z = \alpha_2 \\ z \quad \text{otherwise} \end{cases}$$

Thus by the definition of σ we obtain the inequality $|M\, z| \leq \sigma(z)$ for each $z \in \mathbb{R}$ and for each $M \in \partial\sigma(z)$. □

Proof. (Proposition 5.6)
We compute the gradient of σ and we get

$$\nabla\sigma(z) = \frac{1}{\sum_i e^{z_i}}[e^{z_1}, \ldots, e^{z_n}]^\top = \frac{e^z}{\sum_i e^{z_i}}.$$

Thanks to the convexity of the exponential function we obtain

$$\max_i z_i = \log(e^{\max_i z_i}) \leq \log(e^{\sum_i z_i}) \leq \log(\sum_i e^{z_i}) = \sigma(z)$$

from the previous inequality we obtain

$$|\nabla\sigma(z)^\top z| = \nabla\sigma(z)^\top z = \frac{\sum_i z_i e^{z_i}}{\sum_i e^{z_i}} \leq \max_i z_i \leq \sigma(z).$$

This shows that $\sigma \in \mathrm{subhom}_1(\mathbb{R}^n_+)$. Thanks to the fact that $\nabla\sigma(z)$ is entrywise positive for each $z \in \mathbb{R}^n_+$, for Remark 3.3, $\sigma \in \text{s-subhom}_1(\mathbb{R}^n_+)$. □

B. Additional Experiment

We initiate by considering the general equation for a SubDEQ

$$z = \mathrm{norm}_{\varphi}(\,\sigma_1(\sigma_2(Wz) + f_\theta(x))\,)\,.$$

As we illustrate in Section 4, we can build several categories of SubDEQ layer by choosing either σ_1 or σ_2 as the nonlinear activation function. The two SubDEQs proposed in Section 6 have $\sigma_1 = \mathrm{Id}$ and $\sigma_2(z)$ as the activation functions, if we want to swap σ_1 and σ_2, so σ_1 be the nonlinear function and σ_2 the identity, we must restrict the hidden weights of our DEQ layers to be positive, since in order to apply Lemma 4.1 $\sigma_2(Wz)$ must be positive in the positive orthant. Two examples of this kind are:

$$\mathrm{norm}_{\|\cdot\|_\infty}(\tanh(|W|\,z + \mathrm{ReLU}, (U\,x + b)) + 1.2),$$

and exploiting the power scaling trick

$$\mathrm{norm}_{\|\cdot\|_\infty}(\tanh(|W|\,z + \mathrm{ReLU}(U\,x + b))^{0.99}),$$

where $|W|$ is meant as the absolute value applied entrywise to the weight matrix W. Both layers are well-posed, as their unnormalized versions are subhomogeneous with a degree of 0.99 and differentiable with an entrywise positive Jacobian. We also implement a third variant of SubDEQ without the normalization using the power scaling trick, with the implicit layer defined as follows

$$\tanh(W\,z)^{0.99} + \mathrm{ReLU}(U\,x + b),$$

We test them on the same dataset used in Section 6.2 using the same training, validation, and test splitting with the same hyperparameter of the SubDEQ models in Section 6.2, in Appendix C we report the misclassification error on the test set.

We also conduct other experiments with the graph neural network architecture. In section Section 6.3 we considered the following architecture

$$\begin{cases} \tilde{Z} = \mathrm{norm}_{\|\cdot\|_\infty}\Big(\tanh\big((1-\alpha)\tilde{A}\tilde{Z}\big) + \alpha f_\theta(X) + 1.2\Big) \\ Z = \mathrm{softmax}(\tilde{Z}) \end{cases} \tag{16}$$

Notice that, the map

$$\tilde{Z} \mapsto (1-\alpha)\tilde{A}\tilde{Z}$$

is 1-subhomogeneous, $\alpha f_\theta(X)$ is a positive translation vector, and $\tanh(\cdot) + 1.2$ is 0.99-storngly subhomogeneous. Thus, $\tanh\big((1-\alpha)\tilde{A}\tilde{Z} + \alpha f_\theta(X)\big) + 1.2$, due to Lemma 4.1, has a subhomogeneity degree of 0.99, since is differentiable with an entrywise positive Jacobian, the following iteration converge

$$\begin{cases} \tilde{Z} = \mathrm{norm}_{\|\cdot\|_\infty}\Big(\tanh\big((1-\alpha)\tilde{A}\tilde{Z} + \alpha f_\theta(X)\big) + 1.2\Big) \\ Z = \mathrm{softmax}(\tilde{Z}) \end{cases} \tag{17}$$

We test also this architecture on the same dataset used in Section 6.3 comparing it with APPNP and the subhomogenoeus version of APPNP. Moreover, we tune the α parameter making a grid search on the value $[0.05, 0.1, 0.3, 0.5, 0.7, 0.9]$. To tune α we fixed the test set as half of the unknown labels, and we trained the models by varying α. For each model, we select the optimal α by choosing the one that achieved the highest accuracy on the validation set, the validation set is the half remaining part of the unknown labels. Subsequently, we trained the model five times using the optimal α, altering the training set by sampling from all observations in the dataset that were not part of the initially fixed test set. After each training session, we measured the accuracy of the test set and reported the mean and standard deviation. The results are reported in the table C, where 17 is APPPNP (Normalized Tanh) 2 and 16 is APPPNP (Normalized Tanh). We tested the analogous architectures for the version without the normalization layer.

C. Additional table

Model	Error %
MNIST (Dense)	
SubDEQ (NormalizedTanh (1.603))	2.088 ± 0.1405 %
SubDEQ (Tanh)	$\mathbf{1.92 \pm 0.102}$ **%**
SubDEQ (NormalizedTanh (1.2))	2.437 ± 0.0788 %
SubDEQ (NormalizedwithPowerscaleTanh)	2.568 ± 0.0495 %
SubDEQ (PowerscaleTanh)	1.964 ± 0.125 %
MNIST (Convolutional)	
SubDEQ (NormalizedTanh (1.603))	1.354 ± 0.98 %
SubDEQ (Tanh)	$\mathbf{0.706 \pm 0.011}$ **%**
SubDEQ (NormalizedTanh (1.2))	1.829 ± 0.0888 %
SubDEQ (NormalizedwithPowerscaleTanh)	1.773 ± 0.1948 %
SubDEQ (PowerscaleTanh)	1.316 ± 0.0459 %
CIFAR-10	
SubDEQ (NormalizedTanh (1.603))	28.364 ± 0.377 %
SubDEQ (Tanh)	$\mathbf{27.946 \pm 1.7564}$ **%**
SubDEQ (NormalizedTanh (1.2))	35.809 ± 0.5953 %
SubDEQ (NormalizedwithPowerscaleTanh)	33.864 ± 0.7469 %
SubDEQ (PowerscaleTanh)	30.871 ± 0.2268 %
SVHN	
SubDEQ (Normalized Tanh (1.603))	$\mathbf{9.3562 \pm 0.2122}$ **%**
SubDEQ (Tanh)	10.3987 ± 0.41296 %
SubDEQ (NormalizedTanh (1.2))	25.3903 ± 3.7394 %
SubDEQ (NormalizedwithPowerscaleTanh)	23.4688 ± 4.7033 %
SubDEQ (PowerscaleTanh)	19.4077 ± 0.1346 %

Table 5. Mean $\pm$ standard deviation of the misclassification error on test set

Dataset	Train set size	Validation set size	Test set size
MNIST	71 %	14.5 %	14.5 %
CIFAR-10	70 %	15 %	15 %
SVHN	50.358 %	23.4235 %	26.2184 %

Table 6. Hold out split proportion

Dataset (% labeled)	Accuracy
Cora citation (7.8 %)	
APPNP	$\mathbf{76.3835 \pm 0.7716\%}$
APPNP (Normalized Tanh)	73.4996 ± 0.952 %
APPNP (Normalized Tanh) 2	68.978 ± 0.6919 %
APPNP (Tanh)	74.621 ± 3.2222 %
APPNP (Tanh) 2	75.3079 ± 3.1284 %
Cora author (7.8%)	
APPNP	69.2128 ± 2.7963 %
APPNP (Normalized Tanh)	$\mathbf{69.5713 \pm 1.3922}$ **%**
APPNP (Normalized Tanh) 2	68.1995 ± 2.2836 %
APPNP (Tanh)	68.6672 ± 2.8515 %
APPNP (Tanh) 2	67.8878 ± 2.7859 %
CiteSeer (7.8 %)	
APPNP	$\mathbf{60.9079 \pm 1.3739}$ **%**
APPNP (Normalized Tanh)	59.5334 ± 1.1373 %
APPNP (Normalized Tanh) 2	59.8739 ± 1.6284 %
APPNP (Tanh)	61.2358 ± 1.2669 %
APPNP (Tanh) 2	60.4035 ± 1.4294 %
DBLP (6.0 %)	
APPNP	89.1939 ± 0.2812 %
APPNP (Normalized Tanh)	89.545 ± 0.0526 %
APPNP (Normalized Tanh) 2	86.9683 ± 0.1177 %
APPNP (Tanh)	89.4138 ± 0.269 %
APPNP (Tanh) 2	$\mathbf{89.6610 \pm 0.212}$ **%**
PubMed (1.2%)	
APPNP	75.9526 ± 0.76875 %
APPNP (Normalized Tanh)	76.9303 ± 0.777 %
APPNP (Normalized Tanh) 2	75.0607 ± 0.70495 %
APPNP (Tanh)	$\mathbf{77.9957 \pm 1.0701}$ **%**
APPNP (Tanh) 2	77.2589 ± 0.8646 %

Table 7. Mean $\pm$ standard deviation accuracy on test set

	Models Hyperparameter			
	MNIST (Dense)	MNIST (Conv)	Cifar-10	SVHN
Number of input channels (x)	-	1	3	3
Number of hidden channels (z)	-	16	48	48
Size of hidden channels (z)	-	28×28	32×32	32×32
Hidden kernel size hidden (z)	-	3×3	3×3	3×3
Input kernel size (x)	-	3×3	3×3	3×3
Dimension of input weight matrix (x)	784×87	-	-	-
Dimension of hidden weight matrix (z)	87×87	-	-	-
Average pooling	-	4×4	8×8	8×8
Epochs	30	40	40	40
Initial learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Learning rate schedule	Cosine Annealing	Cosine Annealing	Cosine Annealing	Cosine Annealing
minimum learning rate	10^{-6}	10^{-5}	10^{-3}	10^{-3}
Weight decay	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Batch size	256	256	128	128

Table 8. Models Hyperparameter