Lipschitz Multiscale Deep Equilibrium Models: A Theoretically Guaranteed and Accelerated Approach

Naoki Sato*, Hideaki Iiduka (Meiji University, Japan) *: naoki310303@gmail.com

Key idea: Restructure MDEQ so that the Lipschitz constant L is **user-controllable**, thereby guaranteeing fixed-point convergence and **up to 4.75× speed-up**.

I. Overview

【Background & Motivation】

▷ **Deep equilibrium models (DEQs)** achieve infinitely deep network representations without stacking layers by exploring fixed points of layer transformations in neural networks.

▷ For each new input, a DEQs solve the following fixed-point problem using some fixed-point solver.

Fixed-point problem (each forward pass)

$$\text{Find } \boldsymbol{z} \in \mathrm{Fix}(f_{\boldsymbol{\theta}}) := \{\boldsymbol{z} \in \mathbb{R}^d : f_{\boldsymbol{\theta}}(\boldsymbol{z}; \boldsymbol{x}) = \boldsymbol{z}\} \qquad (1)$$

【Notation】
- $\boldsymbol{z}$: hidden states
- $\boldsymbol{x}$: input image
- $\boldsymbol{\theta}$: weight parameter
- $f_{\boldsymbol{\theta}}$: model / repeated transformation

【Fixed-point solver (example)】
- Banach fixed-point approximation $\boldsymbol{z}_{t+1} := f_{\boldsymbol{\theta}}(\boldsymbol{z}_t)$
- Stopping criterion: $\|f_{\boldsymbol{\theta}}(\boldsymbol{z}_t) - \boldsymbol{z}_t\| \leq \epsilon := 1\text{e-}3$ or $t = 20$

▷ Compared to explicit models, DEQs offers comparable state-of-the-art performance and significantly lower memory usage, but training and inference are **very slow**.

▷ Our goal is to accelerate DEQs training and inference. We focus on MDEQ for the CV task.

Explicit Models (Standard Model)	Counterpart Implicit Models	
Transformer (NeurIPS2017) SSM (ICLR2022)	DEQ-Transformer (NeurIPS2019) Implicit SSM (ICML2025)	NLP task
HRNet (TPAMI2020)	MDEQ (NeurIPS2020)	CV task

【Why MDEQ is slow】

▷ There is no guarantee that Problem (1) has a solution, nor is convergence guaranteed, resulting in high computational cost and **slow** execution.

【Our approach】

▷ From **Banach fixed-point theorem**, a unique solution to problem (1) is guaranteed to exist only if the mapping $f_{\boldsymbol{\theta}}$ is a contraction mapping.

A mapping $f\colon \mathbb{R}^d \to \mathbb{R}^d$ is said to be a contraction mapping if there exists $r \in (0,1)$ such that, for all $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^d$,
$$\|f(\boldsymbol{z}_1) - f(\boldsymbol{z}_2)\| \leq r\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|.$$
That is, a contraction mapping is equivalent to an L-Lipschitz mapping for $L < 1$.
If a mapping f is a contraction mapping, then the sequence generated by the Banach fixed-point approximation method converges to a unique fixed point of f.

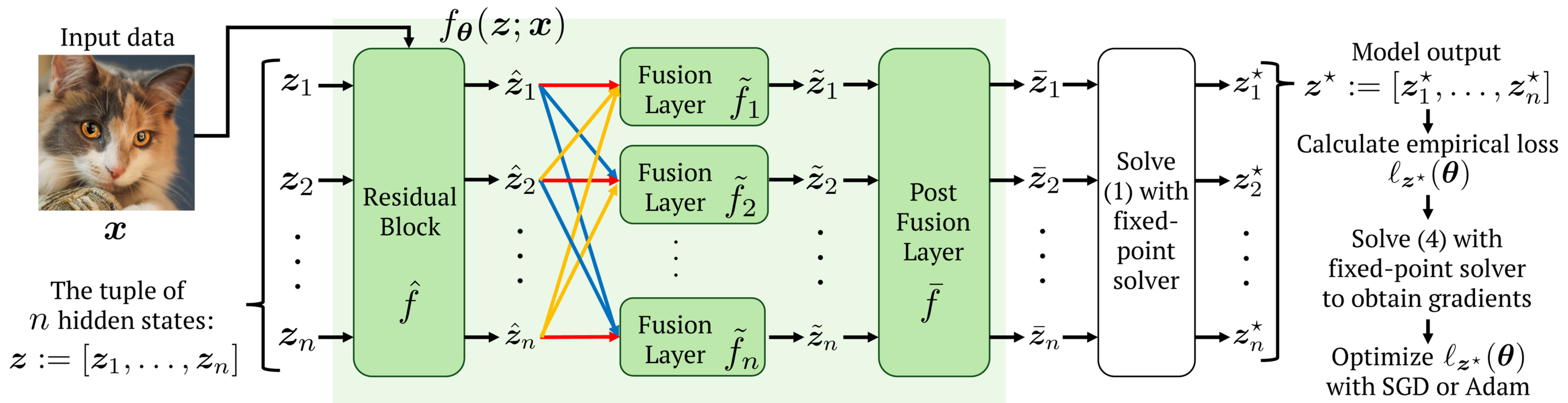
▷ We redesign the model architecture (i.e., $f_{\boldsymbol{\theta}}$) so that there exists a unique solution to the fixed-point problem (1). In other words, modify the model so that the Lipschitz constant of $f_{\boldsymbol{\theta}}$ is less than 1.

【Contribution】

▷ We propose Lipschitz MDEQ, a new variant of MDEQ. This model guarantees fixed-point convergence, thereby enabling accelerated training and inference.

II. Lipschitz MDEQ (ours)

【MDEQ and Lipschitz MDEQ architecture】



MDEQ Architecture [Bai et al., 2020]

$$\begin{cases} \hat{\boldsymbol{z}}_i = \hat{f}_i(\boldsymbol{z}_i) := \mathrm{GN}(\mathrm{ReLU}(\boldsymbol{z}_i + \mathrm{GN}(\mathrm{Drop}(\mathrm{Conv}(\mathrm{ReLU}(\mathrm{GN}(\mathrm{Conv}(\boldsymbol{z}_i)))))))) & \text{(M1)} \\ \tilde{\boldsymbol{z}}_i = \tilde{f}_i(\hat{\boldsymbol{z}}_i) := \hat{\boldsymbol{z}}_i + \sum_{j=1}^{i-1} \mathrm{D}_{i,j}(\hat{\boldsymbol{z}}_i) + \sum_{j=i+1}^{n} \mathrm{U}_{i,j}(\hat{\boldsymbol{z}}_i) & \text{(M2)} \\ \bar{\boldsymbol{z}}_i = \bar{f}_i(\tilde{\boldsymbol{z}}_i) := \mathrm{GN}(\mathrm{Conv}(\mathrm{ReLU}(\tilde{\boldsymbol{z}}_i))) & \text{(M3)} \end{cases}$$

Lipschitz-MDEQ Architecture (ours)

$$\begin{cases} \hat{\boldsymbol{z}}_i = \hat{f}_i(\boldsymbol{z}_i) := \mathrm{MGN}(\mathrm{SReLU}((1-\alpha_1)\boldsymbol{z}_i + \alpha_1 \mathrm{MGN}(\mathrm{Drop}(\mathrm{Conv}^\star(\mathrm{SReLU}(\mathrm{MGN}(\mathrm{Conv}^\star(\boldsymbol{z}_i)))))))) & \text{(L1)} \\ \tilde{\boldsymbol{z}}_i = \tilde{f}_i(\hat{\boldsymbol{z}}_i) := (1-\alpha_2)\hat{\boldsymbol{z}}_i + \alpha_2 \left(\sum_{j=1}^{i-1} w_{ij} \mathrm{D}_{i,j}(\hat{\boldsymbol{z}}_i) + \sum_{j=i+1}^{n} w_{ij} \mathrm{U}_{i,j}(\hat{\boldsymbol{z}}_i) \right) & \text{(L2)} \\ \bar{\boldsymbol{z}}_i = \bar{f}_i(\tilde{\boldsymbol{z}}_i) := \mathrm{MGN}(\mathrm{Conv}^\star(\mathrm{SReLU}(\tilde{\boldsymbol{z}}_i))) & \text{(L3)} \end{cases}$$

Figure 1: Architecture of MDEQ and Lipschitz MDEQ. Lipschitz MDEQ inherits the general structure of MDEQ, with each operation modified to prevent the Lipschitz constant from becoming too large. The sample image is from the AFHQ dataset.

▷ Redesign each operation to have a controllable Lipschitz constant, then combine them so the whole $f_{\boldsymbol{\theta}}$ satisfies $L < 1$.

【Operation modifications (some examples)】

▷ **Activation function**

<ReLU function>
$$\mathrm{ReLU}(\boldsymbol{z})_i := \max\{0, z_i\}$$
Lipschitz constant: $L_{\mathrm{ReLU}} = 1$

⇒ <Scaled-ReLU function>
$$\mathrm{SReLU}(\boldsymbol{z})_i := \max\{0, a z_i\} \quad (a \in (0,1])$$
Lipschitz constant: $L_{\mathrm{SReLU}} = a$

▷ **Normalization**

<Group Normalization>
$$\mathrm{GN}(\boldsymbol{z})_i = \gamma \frac{z_i - \mu_{\boldsymbol{z}}}{\sqrt{\sigma_{\boldsymbol{z}}^2 + \epsilon}} + \beta$$
Lipschitz constant: $L_{\mathrm{GN}} \leq \frac{|\gamma|}{\sqrt{\epsilon}}$

⇒ <Mean-only Group Normalization>
$$\mathrm{MGN}(\boldsymbol{z})_i = \gamma(z_i - \mu_{\boldsymbol{z}}) + \beta$$
Lipschitz constant: $L_{\mathrm{MGN}} \leq |\gamma|$

▷ **Residual connection**
$$h(\boldsymbol{z}) := \boldsymbol{z} + f(\boldsymbol{z})$$
Lipschitz constant: $L_h = 1 + L_f$

⇒ ▷ **Convex Combination**
$$h(\boldsymbol{z}) := (1-\alpha)\boldsymbol{z} + \alpha f(\boldsymbol{z})$$
Lipschitz constant: $L_{h'} = (1-\alpha) + \alpha L_f$

【Lipschitz constant of Lipschitz MDEQ】

Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be L_f-Lipschitz and $g\colon \mathbb{R}^d \to \mathbb{R}^d$ be L_g-Lipschitz.
Then, $f + g$ is an $(L_f + L_g)$-Lipschitz and $f \circ g$ is an $L_f L_g$-Lipschitz.

▷ Therefore, the Lipschitz constant of $f_{\boldsymbol{\theta}}$ is as follows:

$$L = \hat{L}\sqrt{\sum_{i \in [n]} \tilde{L}_i^2}\,\bar{L} \quad \text{under} \quad \begin{cases} \hat{L} = (1-\alpha_1)L_{\mathrm{MGN}}L_{\mathrm{SReLU}} + \alpha_1 L_{\mathrm{MGN}}^3 L_{\mathrm{Conv}^\star}^2 L_{\mathrm{SReLU}}^2 L_{\mathrm{Drop}} \\ \tilde{L}_i = \sqrt{(1-\alpha_2)^2 + \alpha_2^2 \sum_{j \neq i} (w_{ij} L_{\mathrm{fuse}}^{i,j})^2} \\ \bar{L} = L_{\mathrm{MGN}} L_{\mathrm{Conv}^\star} L_{\mathrm{SReLU}} \end{cases}$$

▷ If we set $L_{\mathrm{SReLU}} = 0.35, L_{\mathrm{Conv}^\star} = 2.0, L_{\mathrm{MGN}} = 1.0, \alpha_1 = 0.5, \alpha_2 = 0.3, n = 4, p = 0.3$, then we have L=0.86, i.e., $f_{\boldsymbol{\theta}}$ is a contraction mapping.

【Beneficial Effects in Backward Pass】

▷ We have now been able to guarantee fixed-point convergence for the forward pass (1).

▷ DEQ also solves a fixed-point problem in the backward pass to approximate the gradient. The backward convergence is guaranteed by its smaller Lipschitz constant relative to the forward pass.

III. Numerical Experiments (CIFAR10)

▷ In all experiments, we control the Lipschitz constant L by fixing $L_{\mathrm{MGN}} = 1.0, L_{\mathrm{Conv}^\star} = 2.0, \alpha_1 = 0.5, \alpha_2 = 0.3, n = 4, p = 0.3$ and varying the $L_{\mathrm{SReLU}} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Specifically, $L = \{0.03, 1.0, 14.43\}$ are obtained when $L_{\mathrm{SReLU}} = \{0.1, 0.4, 1.0\}$.

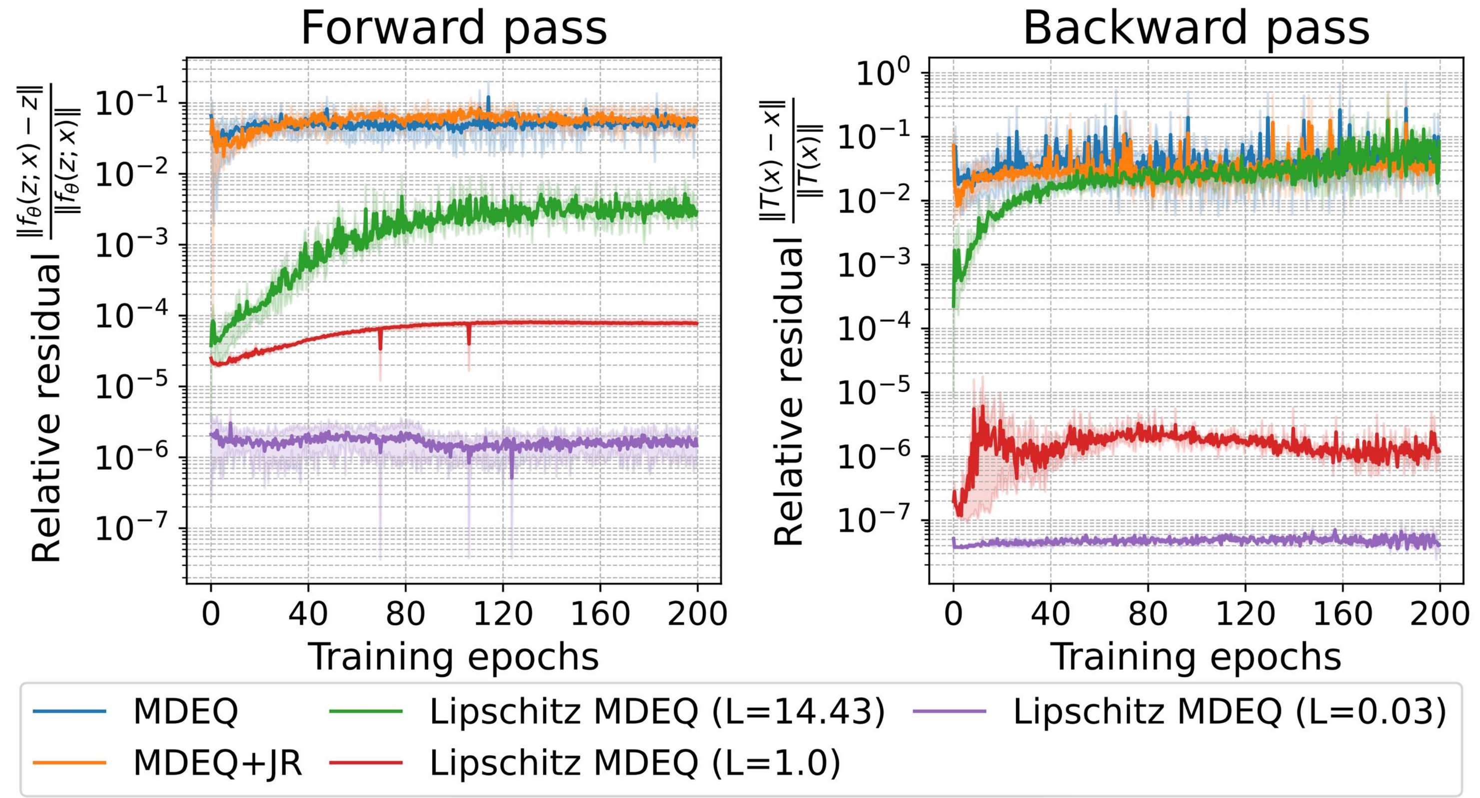
Table 1: Acceleration in Image Classification Tasks

Model and method	Size	Speed-up	Acc.	Train forward		Train backward		Test forward	
				NFEs	Time (ms)	NFEs	Time (ms)	NFEs	Time (ms)
MDEQ	10M	1.0×	93.23	17.9	217	20	291	18.0	63.9
MDEQ + JR	10M	2.22×	91.86	7.0†	74.8	8.0†	97.8	7.0†	39.2
MDEQ + PG	10M	1.41×	92.92	15.6	220	–	87.6	18.0	63.9
MDEQ + JFB	10M	2.84×	88.13	12.3	142	–	**30.6**	11.3	44.3
Lipschitz MDEQ ($L = 14.43$)	10M	1.28×	**93.73**	16.0	187	19.9	193	15.5	54.1
Lipschitz MDEQ ($L = 1.0$)	10M	3.33×	91.44	5.4	75.7	2.9	52.9	4.8	22.6
Lipschitz MDEQ ($L = 0.03$)	10M	**4.75×**	90.47	**2.0**	**44.0**	**2.0**	42.1	**2.0**	**13.3**

▷ Lipschitz MDEQ achieves significantly faster fixed-point convergence than existing methods, reducing runtime.

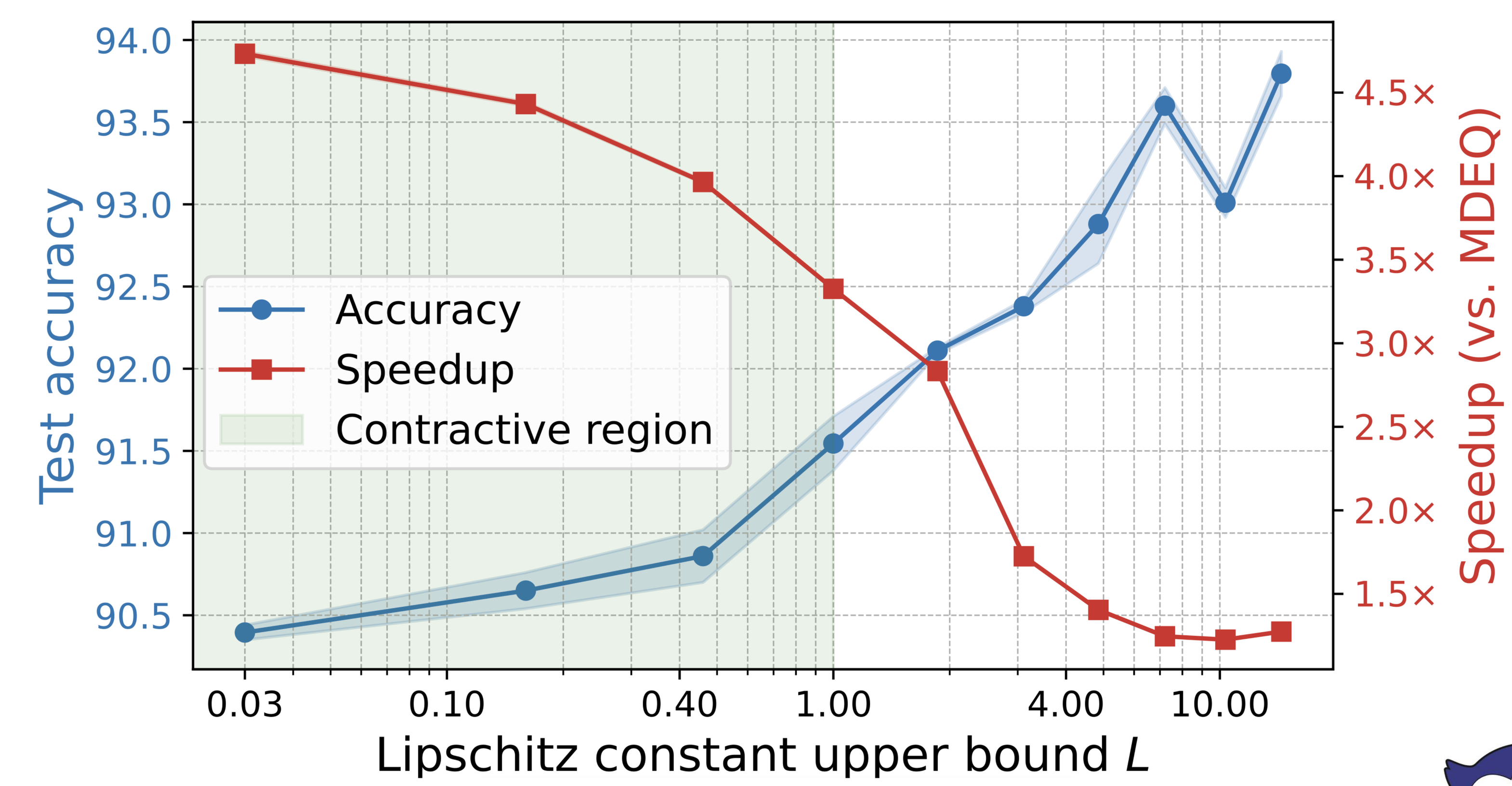
▷ The stricter the contractive constraints, the more test accuracy deteriorated.

Figure 3: Comparison of Fixed-point convergence



▷ Lipschitz MDEQ achieves overwhelming fixed-point convergence, demonstrating better convergence than existing methods even in cases where convergence is is not theoretically guaranteed (i.e., $L \geq 1$).

Figure 4: Trade-off between Accuracy and Runtime



▷ Controlling the Lipschitz constant allows the trade-off between accuracy and speedup to be managed.