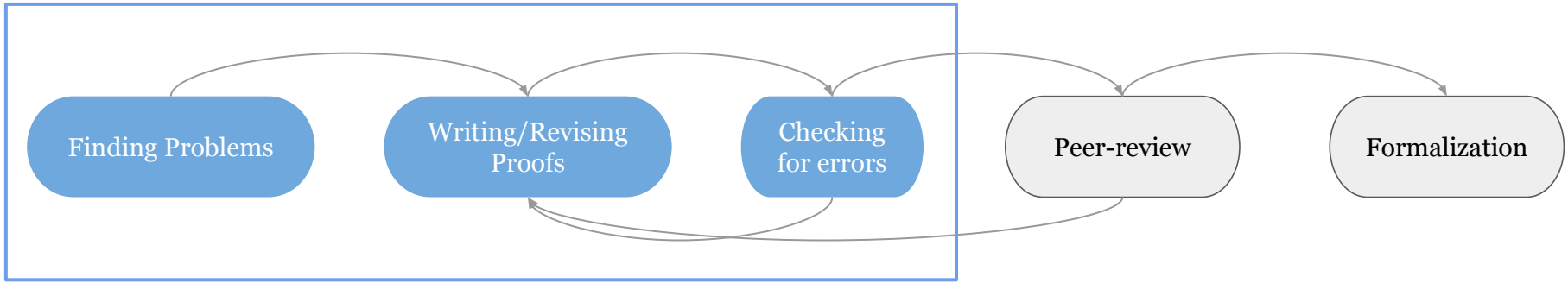
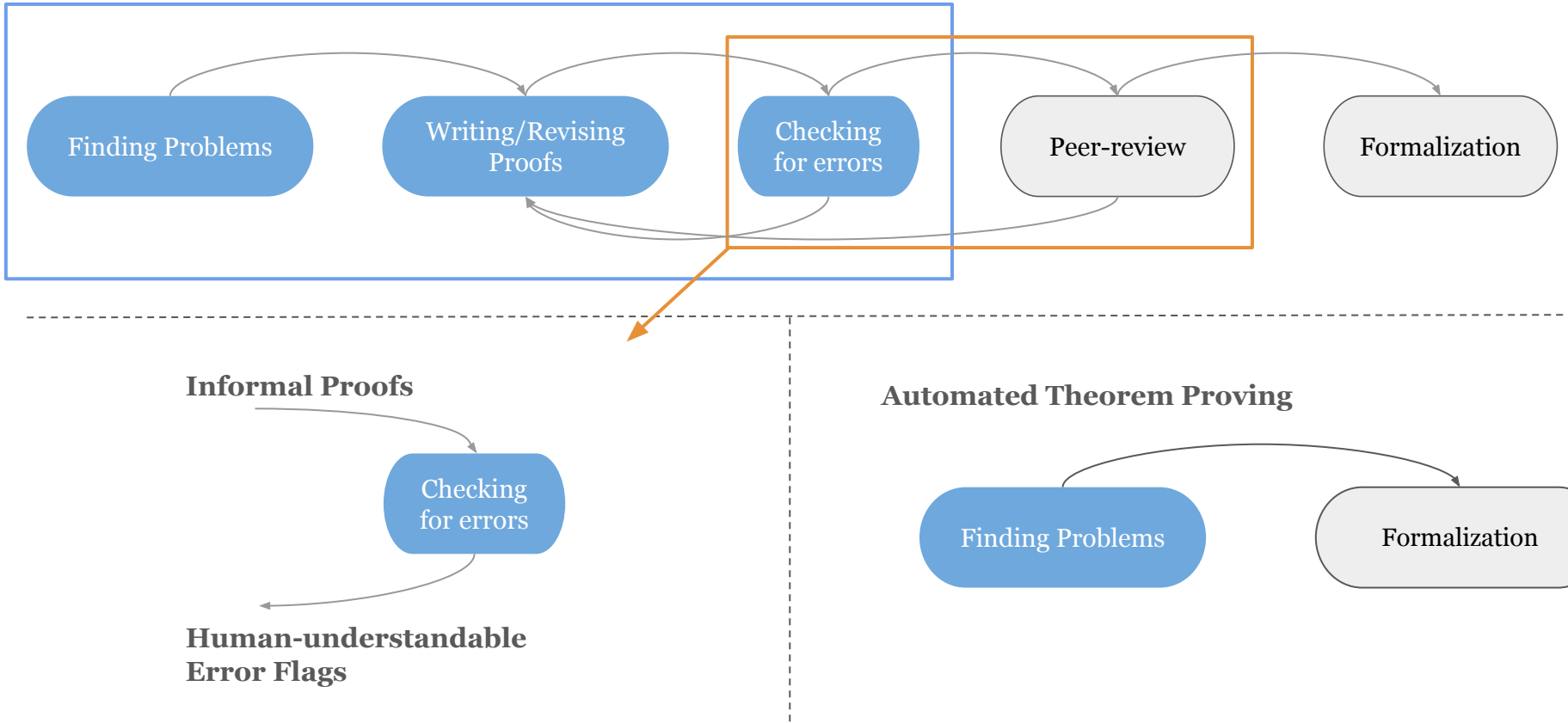
Agent-Based Auditing of Mathematical Proofs in Research Papers

Hieu N. Nguyen & Rui Zhang
The Pennsylvania State University

Building rigorous mathematics

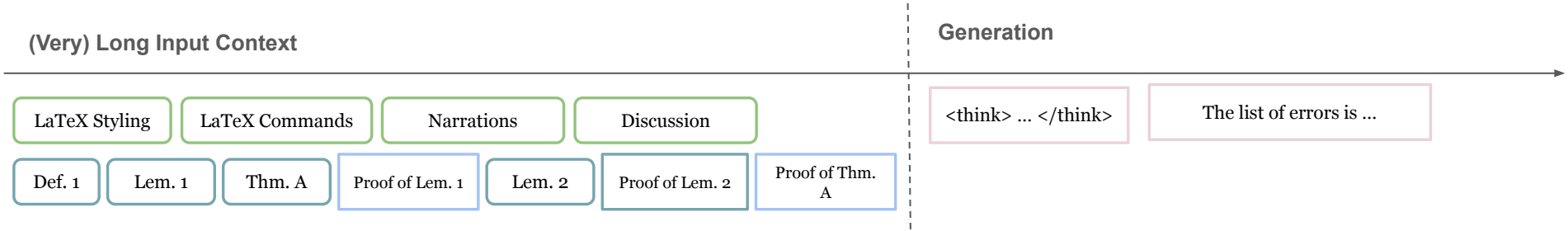


Building rigorous mathematics



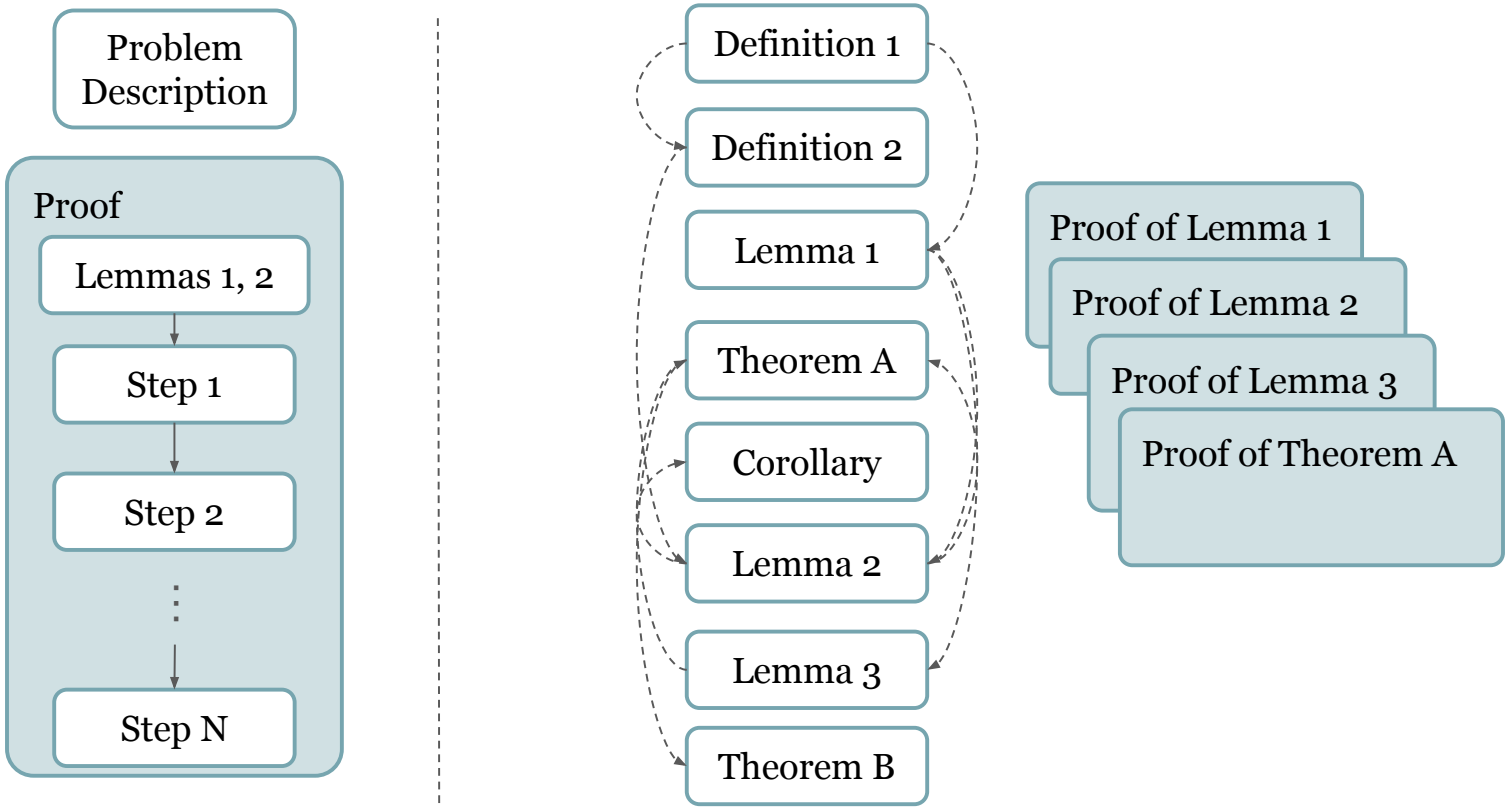
LLM-as-Judge Auditing

A single pass through all scattered components



Can we organize the context to improve performance and efficiency?

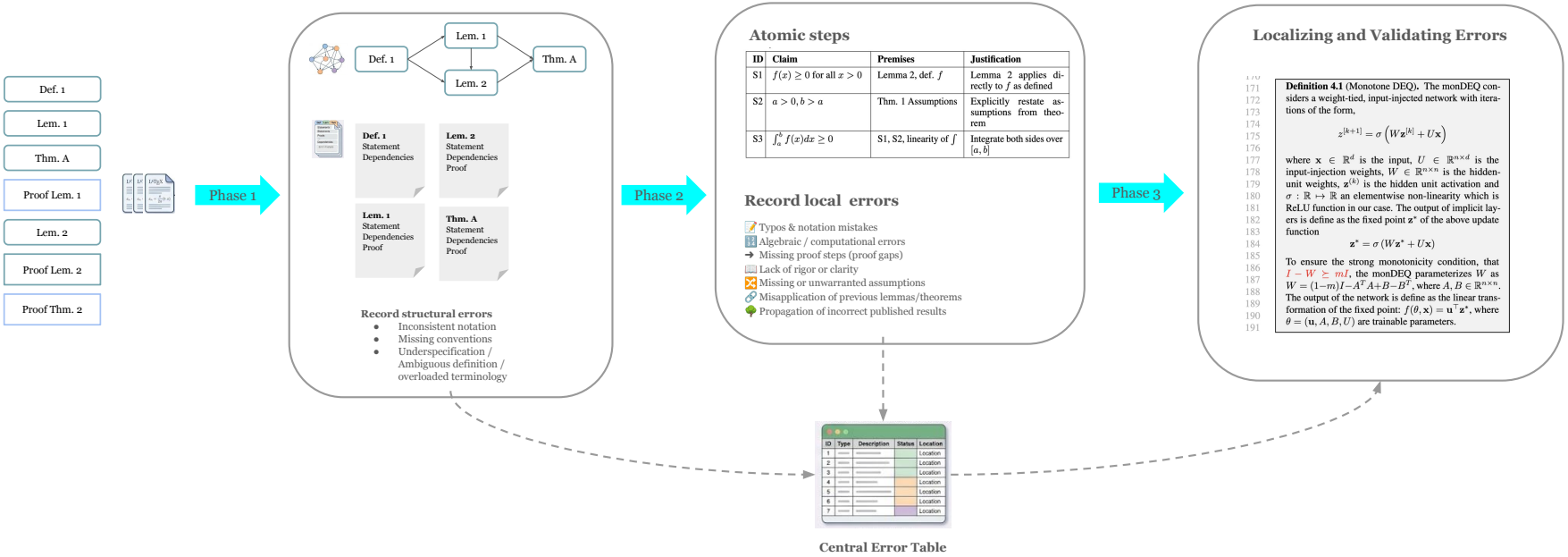
Olympiad-Style Proofs vs. Research-Level Proofs



[1] Zhang, Y. et al (2024) American Invitational Mathematics Examination (AIME) 2024.
[2] Luong, T. et al. (2025) 'Towards robust mathematical reasoning', in Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing.

Agentic Auditing

Phase 1: Identifying Mathematical Components
Phase 2: Auditing Mathematical Components
Phase 3: Error Validation

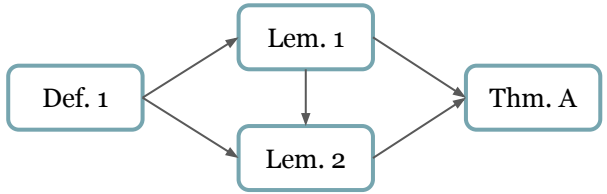


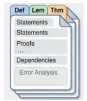
Agentic Auditing

Phase 1: Identifying Mathematical Components

Presentation Order

- Def. 1
- Lem. 1
- Thm. A
- Proof Lem. 1
- Lem. 2
- Proof Lem. 2
- Proof Thm. 2



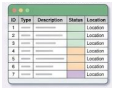


Def. 1
Statement
Dependencies

Lem. 1
Statement
Dependencies
Proof

Lem. 2
Statement
Dependencies
Proof

Thm. A
Statement
Dependencies
Proof



Record structural errors

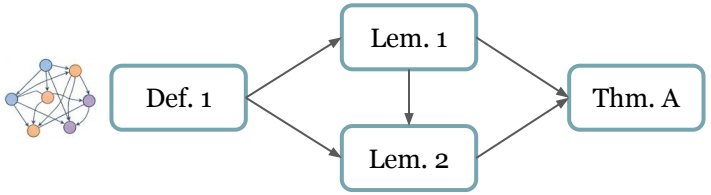
- Inconsistent notation
- Missing conventions
- Underspecification / Ambiguous definitions / Overloaded terminology

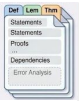
Agentic Auditing

Phase 2: Auditing Mathematical Components

Run for each component following the dependency order

1. Convert the original proof to atomic steps





Thm. A

Given $b > a > 0$, we have $\int_a^b f(x)dx \geq 0$

Proof:

By Lemma 2 and the definition of f, we have f(x) ≥ 0 for all x > 0. Thus, by integrating the inequality over [a,b], we get $\int_a^b f(x)dx \geq 0$

Atomic steps

ID	Claim	Premises	Justification
S1	$f(x) \geq 0$ for all $x > 0$	Lemma 2, def. f	Lemma 2 applies directly to f as defined
S2	$a > 0, b > a$	Thm. A Assumptions	Explicitly restate assumptions from theorem
S3	$\int_a^b f(x)dx \geq 0$	S1, S2	By monotonicity of integration

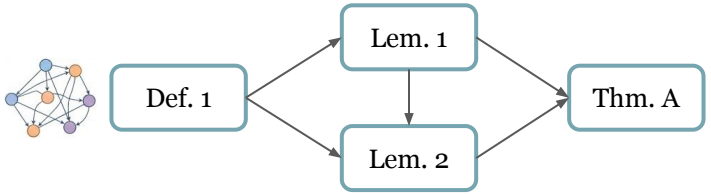
Agentic Auditing

Phase 2: Auditing Mathematical Components

Run for each component following the dependency order

1. Convert the original proof to atomic steps
2. Identify and record local errors

- 📝 Typos & notation mistakes
- 🔢 Algebraic / computational errors
- ➜ Missing proof steps (proof gaps)
- 📖 Lack of rigor or clarity
- 🔀 Missing or unwarranted assumptions
- 🔗 Misapplication of prior results
- 🌳 Propagation of incorrect published results



Atomic steps

ID	Claim	Premises	Justification
S1	$f(x) \geq 0$ for all $x > 0$	Lemma 2, def. f	Lemma 2 applies directly to f as defined
S2	$a > 0, b > a$	Thm. 1 Assumptions	Explicitly restate assumptions from theorem
S3	$\int_a^b f(x)dx \geq 0$	S1, S2, linearity of $\int$	Integrate both sides over $[a, b]$

Agentic Auditing

Phase 3: Error Validation

Central Error Table

ID	Full Error Description
E01	"$I - W \succeq mI$" written as PSD, but W has skew part $B - B^\top$ so $I - W$ is non-symmetric; intended condition is operator strong-monotonicity (symmetrized).
E03	"Contractive for $\alpha \in (0, 2m/L^2]$" — at $\alpha = 2m/L^2$, $L[T_\alpha] \leq 1$ but not strictly < 1.
E04a	L'Hôpital step substitutes $\alpha = 0$ into the unsimplified derivative ratio; the cleaner version takes the limit of the simplified ratio. End answer $1/m$ is correct.

Confirmed

Definition 4.1 (Monotone DEQ). The monDEQ considers a weight-tied, input-injected network with iterations of the form,

$$z^{[k+1]} = \sigma\left(W\mathbf{z}^{[k]} + U\mathbf{x}\right)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input, $U \in \mathbb{R}^{n\times d}$ is the input-injection weights, $W \in \mathbb{R}^{n\times n}$ is the hidden-unit weights, $\mathbf{z}^{(k)}$ is the hidden unit activation and $\sigma : \mathbb{R} \mapsto \mathbb{R}$ an elementwise non-linearity which is ReLU function in our case. The output of implicit layers is define as the fixed point $\mathbf{z}^*$ of the above update function

$$\mathbf{z}^* = \sigma\left(W\mathbf{z}^* + U\mathbf{x}\right)$$

To ensure the strong monotonicity condition, that $I - W \succeq mI$, the monDEQ parameterizes W as $W = (1-m)I - A^TA + B - B^T$, where $A, B \in \mathbb{R}^{n\times n}$. The output of the network is define as the linear transformation of the fixed point: $f(\theta, \mathbf{x}) = \mathbf{u}^\top\mathbf{z}^*$, where $\theta = (\mathbf{u}, A, B, U)$ are trainable parameters.

For each error, trace back to the original document to

1. Identify the location of the error
2. Confirm or refute the flagged issue

Implementation and Experimental Results

Claude Code with a single SKILL.md file.

Run directly on a latex project directory [1]

On Global Convergence Theory for Monotone DEQ

Anonymous Name
Anonymous University

Abstract

Deep Equilibrium (DEQ) models are a class of deep learning models that directly compute the "infinite-depth" feature representation by solving fixed-point equations. Although previous works have experimentally demonstrated the potentials of DEQs in terms of performance and memory usage, the theoretical understanding of training dynamics under gradient descent of DEQ models is still limited. Current theoretical studies either consider deep *linear* equilibrium models, which are a simplified and constrained version of DEQ. In this work, we analyze a more expressive class, called *monotone deep equilibrium (monDEQ)* models. The key difference of monDEQs from linear DEQ models is the monotone operator parametrizations of weight matrices, which introduce nonlinearities, while still guaranteeing the existence of fixed points. In this project, we approach the training convergence problem of monDEQs from the perspective of overparameterization We theoretically prove that under suitable conditions, the gradient descent converges to the global optimum at a linear rate.

1 Introduction

Implicit deep learning models. Recently, a new class of deep learning models, called implicit layers, has been developed based on implicit prediction rules. Instead of having a concrete computation graph, these models define a layer in terms of satisfying joint conditions of input and output. For example, the output z is a fixed point of $f(\cdot, x)$, i.e, $z = f(z, x)$. This formalization leads to deep equilibrium (DEQ) models (Bai et al., 2019). Another examples are differential equations $\frac{dz(t)}{dt} = f(z(t), t)$, leading to Neural ODEs (NODEs) (Chen et al., 2018); and the optimality conditions of optimization problems, leading to differentiable optimization $z(x) = \arg\min_{z \in \mathcal{C}(x)} f(z, x)$ approaches (Amos and Kolter, 2017).

The theoretical question. Neural network-based machine learning is well-known for being somewhat of a "black magic." Its success relies on many tricks; for example, parameter tuning can be quite an art. The conventional belief was that the loss landscape of neural networks with more than two layers is non-convex, highly nonlinear, and high-dimensional. So it was not trivial at all to discover that deep neural networks (DNNs) can be trained. Understanding why first-order algorithms like gradient descent (GD) based methods can often find a global optimum in optimizing the non-convex loss surface of neural networks is one of the critical problems in deep learning theory. This work aims to take a step toward answering the above question for DEQs, a class of architecture that differs from those commonly used in the related literature.

> Under which conditions will training DEQ models with GD converge?

The theoretical Gap. While training dynamics of certain types of DEQ models are understood (for instance, Kawaguchi (2021) have studies linear DEQ models), the dynamics of the more expressive Monotone DEQs (monDEQ) (Winston and Kolter, 2020) have remained a "black box".

In contrast to prior work, we develop a convergence theory for monotone deep equilibrium models (monDEQs) that leverages their operator structure. We prove global linear convergence under gradient descent by leveraging over-parameterization.

2 Related Work

Deep Equilibrium Models. Deep equilibrium models (DEQs), introduced by Bai et al. (2019), define network outputs as the fixed point of a nonlinear transformation. This formulation enables constant-memory training via implicit differentiation and has been applied across sequence modeling, vision, and scientific machine learning. Subsequent work has studied stability, expressivity, and numerical solvers for DEQs, but theoretical understanding of their optimization landscape remains limited.

Global Convergence of Overparameterized DEQs The most closely related work are Kawaguchi (2016); Ling et al. (2022), which establishes that gradient descent converges to a global optimum at a linear rate for overparameterized (linear) DEQs. Their analysis follows the neural tangent kernel (NTK) framework, showing that training dynamics can be approximated by a linear model around initialization. This result relies on a restricted paragrameters where the forward is guareented to be a shrinkage operator. While this line of work provides the first global convergence guarantees for DEQs, we provide a theoretical results for a more expressive class of DEQ models.

Monotone Operator Perspectives in Deep Learning Monotone operator theory has been increasingly used to analyze implicit models and equilibrium networks. Prior work has explored connections between DEQs and monotone operators for ensuring existence and uniqueness of equilibria, as well as for improving numerical stability. However, these approaches have primarily focused on forward stability and solver convergence rather than optimization dynamics.

3 Preliminaries

In this section, we present the definition of the MonDEQ model, the computational procedure, and a lemma that gives an upper bound on the Lipschitz constant of this network. The notation used in this paper is given in Table 1.

In this work, we also use the following two Kronecker identities repeatedly:

$$\mathrm{vec}[ABC] = (C^\top \otimes A)\,\mathrm{vec}[B], \tag{K1}$$

$$\|A \otimes B\|_2 = \|A\|_2 \|B\|_2. \tag{K2}$$

3.1 Setting: Optimizing Monotone DEQ models by Gradient Descent

In this work, we consider a fully connected monotone DEQ ReLU activated network with n hidden neurons in the hidden layers, defined as in Definition 3.1.

Definition 3.1 (Monotone DEQ)**.** *The monDEQ consider a weight-tied, input injected network with iterations of the form,*

$$z^{[k+1]} = \sigma\left(Wz^{[k]} + U\mathbf{x}\right) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input, $U \in \mathbb{R}^{n\times d}$ is the input-injection weights, $W \in \mathbb{R}^{n\times n}$ is the hidden-unit weights, $\mathbf{z}^{(k)}$ is the hidden unit activation and $\sigma : \mathbb{R} \mapsto \mathbb{R}$ an elementwise non-linearity which is ReLU function in our case. The output of implicit layers is define as the fixed point $\mathbf{z}^*$ of the above update function

$$\mathbf{z}^* = \sigma\left(W\mathbf{z}^* + U\mathbf{x}\right) \tag{2}$$

To ensure the strong monotonicity condition, that $I - W \succeq mI$, the monDEQ parameterizes W as

$$W = (1-m)I - A^TA + B - B^T. \tag{3}$$

where $A, B \in \mathbb{R}^{n\times n}$. The output of the network is define as the linear transformation of the fixed point:

$$f(\theta, \mathbf{x}) = \mathbf{u}^\top \mathbf{z}^*$$

where $\theta = (\mathbf{u}, A, B, U)$ are trainable parameters.

An important property of monDEQ is **well-posedness**, i.e., for each input, there exists a unique corresponding output. This is achieved through the parameterization $W = (1-m)I - A^TA + B - B^T$, where the details are provided in Appendix Section B.

Next, we define our input dataset as being drawn from an underlying data distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$. In particular, we draw i.i.d N input-label samples denoted $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. We also denote $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times N}$ and $\mathbf{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$. The equilibrium feature matrix $Z^* \in \mathbb{R}^{n\times N}$ satisfies

$$Z^* = \sigma(WZ^* + UX) \tag{4}$$

where σ applies entry-wise. The prediction vector is

$$\hat{\mathbf{y}} = Z^{*\top}\mathbf{u} \in \mathbb{R}^N.$$

Given a parameterized monDEQ network and the training data S, our optimization problem is the quadratic loss function

$$\min_\theta \mathcal{L}(\theta) = \frac{1}{2}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 = \frac{1}{2}\|Z^{*\top}\mathbf{u} - \mathbf{y}\|_2^2.$$

We train the model by gradient descent (GD) and consider the GD update

$$\theta_{k+1} = \theta_k - \eta\nabla\mathcal{L}(\theta_k),$$

where $\eta > 0$ is the learning rate and $\theta_k = \mathrm{vec}[A_k, B_k, U_k, \mathbf{u}_k]$ is the paraneter we optimize over at step k. The gradient is computed using the implicit function theorem (detailed in the Appendix Section A). Throughout this study, we use k as the GD iteration number, and also use k to index all variables and outputs that depend on θ_k.

3.2 Forward-backward splitting and Bound on $\|Z^*\|_F$

To compute Z^* in practice, we use the forward-backward splitting (FBS) iteration (Winston and Kolter, 2020).

Definition 3.2 (FBS iteration)**.** *Given parameters (W, U) and data X, initialize $Z^{[0]}$ (e.g., to zero) and iterate*

$$Z^{[t+1]} = \sigma\left(T_\alpha Z^{[t]} + \alpha UX\right),$$

where $T_\alpha = I - \alpha(I - W)$ and $\alpha > 0$ is the FBS step size.

Note that we use $\cdot_k$ for GD steps and $\cdot^{[t]}$ for forward-backward splitting steps.

Denote $L[T_\alpha]$ the Lipschitz constant of T_α, we have the following proposition.

Proposition 1 ((Winston and Kolter, 2020))**.** *Let $I - W$ be m-strongly monotone with Lipschitz constant $L = \|I - W\|_2$. Then $L[T_\alpha] \le \sqrt{1 - 2\alpha m + \alpha^2 L^2}$*

This implies that for $\alpha \in \left(0, \frac{2m}{L^2}\right)$ then $L[T_\alpha] < 1$. Thus, the operator T_α is contractive for any $\alpha \in (0, \frac{2m}{L^2})$, and this iteration is guaranteed to converge as long as the operator $I - W$ is Lipschitz and strongly monotone with parameters L.

Lemma 3.1 (Bound on $\|Z^*\|_F$)**.**

$$\|Z^*\|_F \le \frac{\|U\|_2\|X\|_F}{m}. \tag{5}$$

Proof. Run FBS from $Z^{[0]} = 0$ with any $\alpha \in (0, 2m/L^2)$. Since ReLU is non-expansive ($\|\sigma(\mathbf{a})\|_F \le \|\mathbf{a}\|_F$ entry-wise):

$$\begin{aligned}\|Z^{[t]}\|_F &= \|\sigma(T_\alpha Z^{[t-1]} + \alpha UX)\|_F \\ &\le \|T_\alpha Z^{[t-1]} + \alpha UX\|_F \\ &\le L[T_\alpha]\|Z^{[t-1]}\|_F + \alpha\|U\|_2\|X\|_F.\end{aligned}$$

Let $c = L[T_\alpha] \in [0, 1)$ and $r = \alpha\|U\|_2\|X\|_F$. Starting from $Z^{[0]} = 0$:

$$\|Z^{[t]}\|_F \le r\sum_{i=0}^{t-1} c^i \le \frac{r}{1-c} = \frac{\alpha\|U\|_2\|X\|_F}{1 - \sqrt{1 - 2\alpha m + \alpha^2 L^2}}.$$

This bound holds for *every* $\alpha \in (0, 2m/L^2)$. Define

$$f(\alpha) = \frac{\alpha}{1 - \sqrt{1 - 2\alpha m + \alpha^2 L^2}}.$$

We claim $\inf_{\alpha\in(0,\,2m/L^2)} f(\alpha) = 1/m$.

Step 1 (limit): As $\alpha \to 0^+$ the ratio is $0/0$; apply L'Hôpital:

$$\begin{aligned}\lim_{\alpha\to 0^+} f(\alpha) &= \lim_{\alpha\to 0^+} \frac{1}{\frac{d}{d\alpha}\left[1 - \sqrt{1 - 2\alpha m + \alpha^2 L^2}\right]} \\ &= \frac{1}{\frac{m - \alpha L^2}{\sqrt{1 - 2\alpha m + \alpha^2 L^2}}\Big|_{\alpha=0}} \\ &= \frac{1}{m}.\end{aligned}$$

Step 2 (f is increasing): Let $h(\alpha) = 1 - 2\alpha m + \alpha^2 L^2$, so $f(\alpha) = \alpha/(1 - \sqrt{h(\alpha)})$. Differentiating and simplifying (using $h'(\alpha) = -2m + 2\alpha L^2$ and $\sqrt{h} < 1$ on the interval $(0, 2m/L^2)$) shows $f'(\alpha) > 0$ for $\alpha \in (0, 2m/L^2)$. Therefore f is strictly increasing, and its infimum over $(0, 2m/L^2)$ equals its limit at 0^+, which is $1/m$.

Since $\|Z^*\|_F \le f(\alpha)\|U\|_2\|X\|_F$ for every valid α and $\inf f = 1/m$, we conclude

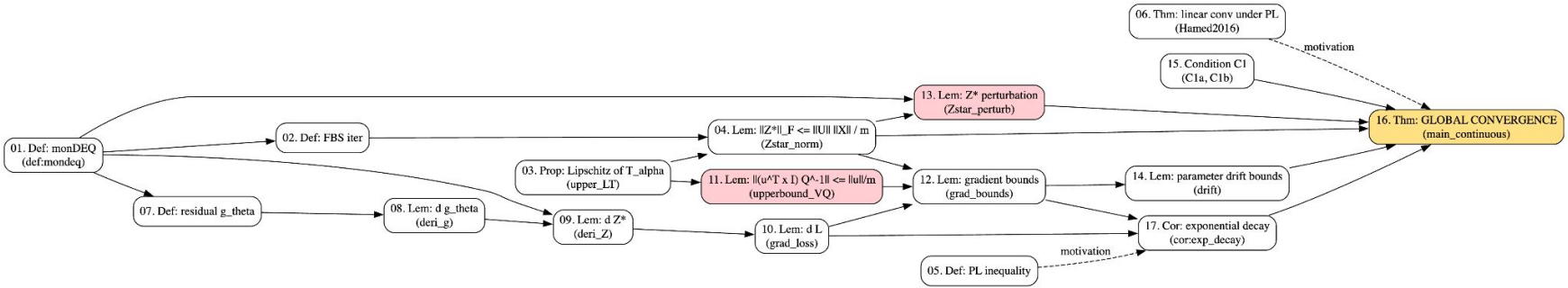
Table 1: Table of Notation

Symbol/E.g	Description
Variables	
Bold lowercase letters $\mathbf{v}$	vectors
Capital letters W	matrices
Operators	
$\langle\cdot,\cdot\rangle$	Inner product
$\|\cdot\|$	Vector norm or operator norm
$\det(\cdot)$	Determinant
$\rho(\cdot)$	Spectral radius
$\lambda_{\max}, \lambda_{\min}(\cdot)$	Max/min eigenvalue
$\sigma_{\max}, \sigma_{\min}(\cdot)$	Max/min singular value
$\otimes$	Kronecker product
$\oplus$	Kronecker sum
$\circ$	Hadamard product
$[n]$	The set $\{1, 2, \ldots, n\}$
$K^{(m)}$	Commutation matrix: $K^{(m)}\mathrm{vec}[A] = \mathrm{vec}[A^\top]$ for $n \times n$ matrices

[1] https://github.com/NNHieu/math_auditing_examples

Implementation and Experimental Results

Dependency Graph



Global Errors

ID	Full Error Description	Severity	Type of Error
S01	vec convention not stated; column-major vs. row-major affects the correctness of exact formulas.	Minor	Ambiguity
S02	“Strong monotonicity” is used ambiguously for both PSD and operator norm interpretations.	Minor	Ambiguity

[1] https://github.com/NNHieu/math_auditing_examples

Implementation and Experimental Results

Local Errors

Definition 4.1 (Monotone DEQ)**.** The monDEQ considers a weight-tied, input-injected network with iterations of the form,

$$z^{[k+1]} = \sigma\left(Wz^{[k]} + Ux\right)$$

where $x \in \mathbb{R}^d$ is the input, $U \in \mathbb{R}^{n\times d}$ is the input-injection weights, $W \in \mathbb{R}^{n\times n}$ is the hidden-unit weights, $z^{(k)}$ is the hidden unit activation and $\sigma : \mathbb{R} \mapsto \mathbb{R}$ an elementwise non-linearity which is ReLU function in our case. The output of implicit layers is define as the fixed point z^* of the above update function

$$z^* = \sigma\left(Wz^* + Ux\right)$$

To ensure the strong monotonicity condition, that $I - W \succeq mI$, the monDEQ parameterizes W as $W = (1-m)I - A^TA + B - B^T$, where $A, B \in \mathbb{R}^{n\times n}$. The output of the network is define as the linear transformation of the fixed point: $f(\theta, x) = u^\top z^*$, where $\theta = (u, A, B, U)$ are trainable parameters.

Proposition 4.2. *Let $I - W$ be m-strongly monotone with Lipschitz constant $L = \|I - W\|_2$. Then*

$$L[T_\alpha] \le \sqrt{1 - 2\alpha m + \alpha^2 L^2}.$$

This implies that for $\alpha \in \left(0, \frac{2m}{L^2}\right)$,

$$L[T_\alpha] < 1.$$

Thus, the operator T_α is contractive for any $\alpha \in (0, \frac{2m}{L^2}]$, and this iteration is guaranteed to converge as long as the operator $I - W$ is Lipschitz and strongly monotone with parameter L.

Lemma 4.3. $\|[I - W^\top J]^{-1}u\| \le \frac{\|u\|}{m}$

Proof. Let $\Pi = [J + \mu(I - J)]^{-1}$ with $\mu > 0$. Since $J_{ii} < 1 \forall i \in [n]$, we have $\Pi \in D^+$.

... Let $G = \{[I + \mu W^\top (I - J)]\Pi - I\}$ and $F(v') = (I - W^\top)v' - u$.

The resolvent operator for G is $R_G = (I + \gamma G)^{-1}$, with $\gamma = 1$ then $R_G = \Pi^{-1}\left[I + \mu W^\top (I - J)\right]^{-1}$. The update of forward-backward splitting is

$$\begin{aligned}\bar{v}^{k+1} =& R_G(\bar{v}^k - \alpha F(\bar{v}^k)) \\ =& \Pi^{-1}\left[I + \mu W^\top (I - J)\right]^{-1} \\ & \{[I - \alpha(I - W^\top)]\bar{v}^k + \alpha u\}\end{aligned}$$

From the proposition 4.2, we have $L[[I - \alpha(I - W^\top)]] \le \sqrt{1 - 2\alpha m + \alpha^2 L[I - W^\top]^2}$. Take the limit $\mu \to 0^+$, we have $\|\Pi\| \le 1$ and $\|R_G\| \le 1$.

Same argument from lemma **??**, we have $\|\bar{v}^*\| \le \frac{\|u\|}{m}$. Further,

$$v^* = \Pi\bar{v}^* = \{[I - \alpha(I - W^\top)]\bar{v}^* + \alpha u\}$$

when $\mu \to 0$). So $\|v^*\| \le \|\bar{v}^*\| = \frac{\|u\|}{m}$. □

- 23 of the 25 detected errors were confirmed
- The LLM-as-a-judge baseline found only 5 errors

[1] https://github.com/NNHieu/math_auditing_examples

Conclusion

In this work, we

1. Highlight the critical need for automated auditing of informal mathematical proofs, and
2. Introduce an agent-based assistant to address this challenge.

Current limitations include:

1. A very small evaluation sample size,
2. Limited evaluation metrics, and
3. Lack of strong baselines and ablation studies.

[1] https://github.com/NNHieu/math_auditing_examples

Thank you

Q & A