1 Table of Notions

The notations in this work are summarized in Tab. 1.

Table 1: Table of notations in this work.

Symbol	Description
	Vectors
$\boldsymbol{x}$	Input data
$\boldsymbol{u}$	Injection of input $\boldsymbol{x}$
$\boldsymbol{h}$	Intermediate feature of input $\boldsymbol{x}$
$\hat{\boldsymbol{y}}$	Prediction of input $\boldsymbol{x}$
$\boldsymbol{y}$	Groundtruth of input $\boldsymbol{x}$
$\boldsymbol{\theta}$	Parameter vector of the equilibrium module
$\boldsymbol{z}$	A union of $\boldsymbol{u}$ and $\boldsymbol{\theta}$
	Functions
$\mathcal{M}(\boldsymbol{x})$	Preprocessing module, $\mathcal{M}:\mathbb{R}^{d_{\boldsymbol{x}}}\to\mathbb{R}^{d_{\boldsymbol{u}}}$
$\mathcal{F}(\boldsymbol{h},\,\boldsymbol{z})$	Equilibrium module, $\mathcal{F}:\mathbb{R}^{d_{\boldsymbol{h}}}\times\mathbb{R}^{d_{\boldsymbol{z}}}\to\mathbb{R}^{d_{\boldsymbol{h}}}$
$\mathcal{G}(\boldsymbol{h})$	Postprocessing module, $\mathcal{G}:\mathbb{R}^{d_{\boldsymbol{h}}}\to\mathbb{R}^{d_{\boldsymbol{y}}}$
$\mathcal{R}(\boldsymbol{\theta})$	Loss function, $\mathcal{R}:\mathbb{R}^{d_{\boldsymbol{\theta}}}\to\mathbb{R}$
	Equilibrium states $\boldsymbol{h}$
$\boldsymbol{h}^*$	The equilibrium point of $\mathcal{F}$ given $\boldsymbol{z}$
$\boldsymbol{h}_t$	The intermediate feature of the t^{th} unrolled step
	Gradients & Jacobians
$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}$	Exact gradient of the loss *w.r.t.* the parameters $\boldsymbol{\theta}$
$\widehat{\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}}$	Phantom gradient, *i.e.,* an approximation to $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}$
$\frac{\partial\boldsymbol{a}}{\partial\boldsymbol{b}}$	Gradient of $\boldsymbol{a}$ *w.r.t.* $\boldsymbol{b}$, *i.e.,* $\left(\frac{\partial\boldsymbol{a}}{\partial\boldsymbol{b}}\right)_{ij}=\frac{\partial\boldsymbol{a}_j}{\partial\boldsymbol{b}_i}$.
$\boldsymbol{A}$	An approximation to $\frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}\left(\boldsymbol{I}-\frac{\partial\mathcal{F}}{\partial\boldsymbol{h}}\right)^{-1}$
$\boldsymbol{D}$	An approximation to $\left(\boldsymbol{I}-\frac{\partial\mathcal{F}}{\partial\boldsymbol{h}}\right)^{-1}$
	Scalars
$\sigma_{\max},\,\sigma_{\min}$	The maximal/minimal singular value of $\frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}$
κ	The condition number of $\frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}$
$k,\,\lambda$	The number of steps and damping factor of phantom gradient
$L_{\boldsymbol{h}}$	The Lipschitz constant of $\mathcal{F}$ *w.r.t.* $\boldsymbol{h}$
$d_\diamond$	Dimension of vector $\diamond$
	Operators
$\langle\cdot,\,\cdot\rangle$	Inner product
$\|\cdot\|$	Vector norm or operator norm
$\rho(\cdot)$	Spectral radius

2 Algorithm of Phantom Gradient

The following PyTorch-style [1] pseudocode describes the implementation of both the unrolling-based phantom gradient (see Alg. 1) and the Neumann-series-based one (see Alg. 2). To implement the phantom gradient with TensorFlow [2], replace the no_grad context manager with the $stop\_gradient$ operator.

The unrolling-based phantom gradient is computed by the automatic differentiation engine, while the Neumann-series-based phantom gradient is given by Alg. 3. A special reminder is that, for a trained model, removing the unrolling steps in the test stage will not lead to a performance decay but accelerate the inference instead. Similarly, increasing the unrolling steps in the test stage can not further improve the performance, which is validated using MDEQ model on the CIFAR-10 and ImageNet datasets. This implies that the root-finding solver has fully converged to an equilibrium point for the trained model.

Algorithm 1 Unrolling-based phantom gradient, PyTorch-style

```
# solver: the solver to find h*, e.g., the Broyden solver in MDEQ.
# func: the explicit function F that defines the implicit model.
# z: the input variables z to solve h* = F(h*, z)
# h: the solution h* of the implicit module.
# k: the unrolling steps k.
# lambda_: the damping factor λ.
# training: a bool variable that indicates the training or inference stage.

# Forward pass (Backward pass is accomplished by automatic differentiation)
def forward(z, k, lambda_, training):
    with torch.no_grad():
        h = solver(func, z)

    if training:
        for _ in range(k):
            h = (1 - lambda_) * h + lambda_ * func(h, z)

    return h
```

3 Proof of Theorems

Theorem 1. *Suppose the exact gradient and the phantom gradient are given by Eq. (4) and (5), respectively. Let σ_{max} and σ_{min} be the maximal and minimal singular value of $\partial\mathcal{F}/\partial\boldsymbol{\theta}$. If*

$$\left\| \boldsymbol{A}\left(\boldsymbol{I}-\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right)-\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right\|<\frac{\sigma_{min}^2}{\sigma_{max}}, \tag{A-1}$$

then the phantom gradient provides an ascent direction of the function $\mathcal{L}$, i.e.,

$$\left\langle \widehat{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}}, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right\rangle>0. \tag{A-2}$$

Proof. Denote $\boldsymbol{J}=\partial\mathcal{F}/\partial\boldsymbol{\theta}$, $\boldsymbol{v}=\partial\mathcal{L}/\partial\boldsymbol{h}$, and $\boldsymbol{u}=\left(\boldsymbol{I}-\partial\mathcal{F}/\partial\boldsymbol{h}\right)^{-1}\boldsymbol{v}$. Let

$$\boldsymbol{E}=\boldsymbol{A}\left(\boldsymbol{I}-\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right)-\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}, \tag{A-3}$$

and we have $\|\boldsymbol{E}\| \leq \sigma_{\min}^2/\sigma_{\max}$. Then,

$$\begin{aligned}\left\langle \widehat{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}}, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right\rangle &= \boldsymbol{v}^{\top}\boldsymbol{A}^{\top}\boldsymbol{J}\left(\boldsymbol{I}-\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right)^{-1}\boldsymbol{v}=\boldsymbol{u}^{\top}\left(\boldsymbol{I}-\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right)^{\top}\boldsymbol{A}^{\top}\boldsymbol{J}\boldsymbol{u}=\boldsymbol{u}^{\top}\left(\boldsymbol{J}+\boldsymbol{E}\right)^{\top}\boldsymbol{J}\boldsymbol{u} \\ &\geq \|\boldsymbol{J}\boldsymbol{u}\|^2-\|\boldsymbol{E}\|\|\boldsymbol{J}\|\|\boldsymbol{u}\|^2 \geq \left(\sigma_{\min}^2-\sigma_{\max}\|\boldsymbol{E}\|\right)\|\boldsymbol{u}\|^2>0,\end{aligned} \tag{A-4}$$

Algorithm 2 Neumann-series-based Phantom Gradient, Pytorch-style

```
# solver: the solver to find h*, e.g., the Broyden solver in MDEQ.
# func: the explicit function F that defines the implicit model.
# grad(a, b, c): the function to compute the Jacobian-vector product
    (∂a/∂b) c
# z: the input variables z to solve h* = F(h*, z)
# h: the output h* of the implicit module.
# g: the input gradient ∂L/∂h.
# g_out: the output gradient ∂L/∂z.
# k: the unrolling steps k.
# lambda_: the damping factor λ.

# Forward pass
def forward(z):
    with torch.no_grad():
        h = solver(func, z)

    return h

# Backward pass
def phantom_grad(g, h, z, k, lambda_):
    f = (1 - lambda_) * h + lambda_ * func(h, z)

    g_hat = g
    for _ in range(k-1):
        g_hat = g + grad(f, h, g_hat)

    g_out = lambda_ * grad(f, z, g_hat)
    return g_out
```

Algorithm 3 Neumann-series-based phantom gradient with $\mathcal{O}(1)$ memory

Input $\partial\mathcal{L}/\partial\boldsymbol{h}$, $\mathcal{F}$, $\boldsymbol{h}^*$, k, λ.
Initialize $\hat{\boldsymbol{g}} = \boldsymbol{g} = \partial\mathcal{L}/\partial\boldsymbol{h}$;
$\boldsymbol{f} \leftarrow (1-\lambda)\boldsymbol{h}^* + \lambda\mathcal{F}(\boldsymbol{h}^*, \boldsymbol{z})$
for $i = 1, 2, \cdots, k-1$ **do**
    $\hat{\boldsymbol{g}} \leftarrow \boldsymbol{g} + (\partial\boldsymbol{f}/\partial\boldsymbol{h})\,\hat{\boldsymbol{g}}$; ▷ Compute Jacobian-vector product with automatic differentiation
end for
$\boldsymbol{g}_{\text{out}} \leftarrow \lambda\,(\partial\boldsymbol{f}/\partial\boldsymbol{z})\,\hat{\boldsymbol{g}}$ ▷ Compute Jacobian-vector product to obtain the phantom gradient *w.r.t.* $\boldsymbol{z}$
return $\hat{\boldsymbol{g}}$.

which concludes the proof. □

Proof of Remerk 1. Suppose $\boldsymbol{A} = (\partial\mathcal{F}/\partial\boldsymbol{\theta})\,\boldsymbol{D}$ and the condition in (8). Then,

$$\left\|\boldsymbol{A}\left(\boldsymbol{I} - \frac{\partial\mathcal{F}}{\partial\boldsymbol{h}}\right) - \frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}\right\| \le \left\|\frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}\right\|\left\|\boldsymbol{D}\left(\boldsymbol{I} - \frac{\partial\mathcal{F}}{\partial\boldsymbol{h}}\right) - \boldsymbol{I}\right\| < \sigma_{\max}\cdot\frac{1}{\kappa^2} = \frac{\sigma_{\min}^2}{\sigma_{\max}}, \tag{A-5}$$

indicating the condition in (A-1) is satisfied. □

Theorem 2. *Suppose the Jacobian $\partial\mathcal{F}/\partial\boldsymbol{h}$ is a contraction mapping. Then,*

(i) the Neumann series in (14) converges to the Jacobian-inverse $(\boldsymbol{I} - \partial\mathcal{F}/\partial\boldsymbol{h})^{-1}$; and

(ii) if the function $\mathcal{F}$ is continuously differentiable w.r.t. both $\boldsymbol{h}$ and $\boldsymbol{\theta}$, the sequence in Eq. (13) converges to the exact Jacobian $\partial\boldsymbol{h}^*/\partial\boldsymbol{\theta}$ as $T \to \infty$, i.e.,

$$\lim_{T\to\infty}\frac{\partial\boldsymbol{h}_T}{\partial\boldsymbol{\theta}} = \left.\frac{\partial\mathcal{F}}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{h}^*}\left(\boldsymbol{I} - \left.\frac{\partial\mathcal{F}}{\partial\boldsymbol{h}}\right|_{\boldsymbol{h}^*}\right)^{-1}. \tag{A-6}$$

Proof. (i) Since $\|\partial\mathcal{F}/\partial\boldsymbol{h}\| < 1$,

$$\|\boldsymbol{B}\| \leq \lambda \left\|\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right\| + (1-\lambda)\|\boldsymbol{I}\| < 1. \tag{A-7}$$

Let $\boldsymbol{B}_k = \sum_{t=0}^{k-1} \boldsymbol{B}^t$, and for each $p \in \mathbb{N}_+$, we have

$$\|\boldsymbol{B}_{k+p} - \boldsymbol{B}_k\| = \left\|\sum_{t=k}^{k+p-1} \boldsymbol{B}^t\right\| \leq \|\boldsymbol{B}\|^k \left\|\sum_{t=0}^{p-1} \boldsymbol{B}^t\right\| \leq \|\boldsymbol{B}\|^k \sum_{t=0}^{p-1} \|\boldsymbol{B}\|^t < \frac{\|\boldsymbol{B}\|^k}{1-\|\boldsymbol{B}\|}. \tag{A-8}$$

By the Cauchy's convergence test, the sequence $\{\boldsymbol{B}_k\}$ is convergent. Since

$$(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{B}_k = \boldsymbol{I} - \boldsymbol{B}^k \to \boldsymbol{I}, \quad \text{as } k \to \infty, \tag{A-9}$$

it follows that $\boldsymbol{B}_k \to (\boldsymbol{I}-\boldsymbol{B})^{-1}$, as $k \to \infty$. Therefore,

$$\lambda \sum_{t=0}^{\infty} \boldsymbol{B}^t = \lambda (\boldsymbol{I}-\boldsymbol{B})^{-1} = \left(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right)^{-1}. \tag{A-10}$$

(ii) Let $\mathcal{F}_\lambda(\boldsymbol{h}, \boldsymbol{z}) = \lambda\mathcal{F}(\boldsymbol{h}, \boldsymbol{z}) + (1-\lambda)\boldsymbol{h}$, and

$$\frac{\partial \mathcal{F}_\lambda}{\partial \boldsymbol{h}} = \lambda \frac{\partial \mathcal{F}}{\partial \boldsymbol{h}} + (1-\lambda)\boldsymbol{I}. \tag{A-11}$$

Similar to (A-7), $\partial\mathcal{F}_\lambda/\partial\boldsymbol{h}$ is also a contraction mapping. By the Banach Fixed Point Theorem [3], the sequence $\{\boldsymbol{h}_t\}$ converges to an exact fixed point $\boldsymbol{h}^*$ of $\mathcal{F}_\lambda$, which is also a fixed point of $\mathcal{F}$.

Denote

$$\boldsymbol{U}_t = \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{h}_t}, \quad \boldsymbol{V}_t = \lambda \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right|_{\boldsymbol{h}_t} + (1-\lambda)\,\boldsymbol{I}. \tag{A-12}$$

Since the function $\mathcal{F}$ is continuously differentiable *w.r.t.* both $\boldsymbol{h}$ and $\boldsymbol{\theta}$, we have

$$\lim_{t\to\infty} \boldsymbol{U}_t = \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{h}^*} = \boldsymbol{U}_\infty, \quad \lim_{t\to\infty} \boldsymbol{V}_t = \lambda \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right|_{\boldsymbol{h}^*} + (1-\lambda)\,\boldsymbol{I} = \boldsymbol{V}_\infty. \tag{A-13}$$

According to the conclusion in (i), we have

$$\left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{h}^*} \left(\boldsymbol{I} - \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right|_{\boldsymbol{h}^*}\right)^{-1} = \lambda\, \boldsymbol{U}_\infty \sum_{t=0}^{\infty} \boldsymbol{V}_\infty^t. \tag{A-14}$$

Comparing Eq. (13) with Eq. (32), we have

$$\begin{aligned}
&\left\|\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{\theta}} - \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{h}^*} \left(\boldsymbol{I} - \left.\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right|_{\boldsymbol{h}^*}\right)^{-1}\right\| = \lambda \left\|\sum_{t=0}^{T-1} \boldsymbol{U}_t \prod_{s=t+1}^{T-1} \boldsymbol{V}_s - \boldsymbol{U}_\infty \sum_{t=0}^{\infty} \boldsymbol{V}_\infty^t\right\| \\
\leq \lambda &\left(\underbrace{\left\|\sum_{t=0}^{T-2} \boldsymbol{U}_t \left(\prod_{s=t+1}^{T-1} \boldsymbol{V}_s - \boldsymbol{V}_\infty^{T-t-1}\right)\right\|}_{\boldsymbol{\Delta}_1} + \underbrace{\left\|\sum_{t=0}^{T-1} (\boldsymbol{U}_t - \boldsymbol{U}_\infty)\, \boldsymbol{V}_\infty^{T-t-1}\right\|}_{\boldsymbol{\Delta}_2} + \underbrace{\left\|\boldsymbol{U}_\infty \sum_{t=T}^{\infty} \boldsymbol{V}_\infty^t\right\|}_{\boldsymbol{\Delta}_3} \right).
\end{aligned} \tag{A-15}$$

In the following context, we prove Eq. (A-6) by showing that $\boldsymbol{\Delta}_1$, $\boldsymbol{\Delta}_2$, and $\boldsymbol{\Delta}_3$ can be arbitrarily small when T is sufficiently large.

Preparations. For any $\epsilon > 0$, since $\boldsymbol{U}_t \to \boldsymbol{U}_\infty$ and $\boldsymbol{V}_t \to \boldsymbol{V}_\infty$ as $t \to \infty$, there exists $N \in \mathbb{N}_+$ s.t.

$$\|\boldsymbol{U}_t - \boldsymbol{U}_\infty\| < \epsilon, \quad \|\boldsymbol{V}_t - \boldsymbol{V}_\infty\| < \epsilon, \quad \forall t > N. \tag{A-16}$$

Since $\partial\mathcal{F}_\lambda/\partial\boldsymbol{h}$ is a contraction mapping, there exists $\gamma \in (0,1)$ s.t.

$$\|\boldsymbol{V}_t\| \leq \gamma, \quad \|\boldsymbol{V}_\infty\| \leq \gamma. \tag{A-17}$$

Besides, since $\partial\mathcal{F}/\partial\boldsymbol{\theta}$ is a continuous function and $\{\boldsymbol{h}_t\}$ is a convergent sequence, it follows that $\{\boldsymbol{h}_t\}$ is contained by a compact set and that $\partial\mathcal{F}/\partial\boldsymbol{\theta}$ is bounded on $\{\boldsymbol{h}_t\}$. Therefore, there exists $M > 0$, s.t.

$$\|\boldsymbol{U}_t\| \leq M, \quad t = 0, 1, 2, \cdots. \tag{A-18}$$

Taking $t \to \infty$, we have $\|\boldsymbol{U}_\infty\| \leq M$.

For $\boldsymbol{\Delta}_1$. For $t > N$, consider

$$\begin{aligned}
&\left\| \boldsymbol{U}_t \left(\prod_{s=t+1}^{T-1} \boldsymbol{V}_s - \boldsymbol{V}_\infty^{T-t-1} \right) \right\| \\
&\leq \|\boldsymbol{U}_t\| \sum_{s=t+1}^{T-1} \left\| \boldsymbol{V}_{t+1}\boldsymbol{V}_{t+2}\cdots\boldsymbol{V}_s\boldsymbol{V}_\infty^{T-s-1} - \boldsymbol{V}_{t+1}\boldsymbol{V}_{t+2}\cdots\boldsymbol{V}_{s-1}\boldsymbol{V}_\infty^{T-s} \right\| \\
&\leq \|\boldsymbol{U}_t\| \sum_{s=t+1}^{T-1} \|\boldsymbol{V}_{t+1}\| \|\boldsymbol{V}_{t+2}\| \cdots \|\boldsymbol{V}_{s-1}\| \|\boldsymbol{V}_s - \boldsymbol{V}_\infty\| \|\boldsymbol{V}_\infty\|^{T-s-1} \\
&\leq M(T-t-1)\gamma^{T-t-2}\epsilon,
\end{aligned} \tag{A-19}$$

and for $t \leq N$, we simply have

$$\left\| \boldsymbol{U}_t \left(\prod_{s=t+1}^{T-1} \boldsymbol{V}_s - \boldsymbol{V}_\infty^{T-t-1} \right) \right\| \leq \|\boldsymbol{U}_t\| \left(\prod_{s=t+1}^{T-1} \|\boldsymbol{V}_s\| + \|\boldsymbol{V}_\infty\|^{T-t-1} \right) \leq 2M\gamma^{T-t-1}. \tag{A-20}$$

Therefore, when $T > N+2$, $\boldsymbol{\Delta}_1$ can be bounded as follows:

$$\begin{aligned}
\boldsymbol{\Delta}_1 &\leq \left(\sum_{t=0}^{N} + \sum_{t=N+1}^{T-2} \right) \left\| \boldsymbol{U}_t \left(\prod_{s=t+1}^{T-1} \boldsymbol{V}_s - \boldsymbol{V}_\infty^{T-t-1} \right) \right\| \\
&\leq 2M \sum_{t=0}^{N} \gamma^{T-t-1} + M\epsilon \sum_{t=N+1}^{T-2} (T-t-1)\gamma^{T-t-2} \\
&\leq 2M\gamma^{T-N-1} \frac{1-\gamma^{N+1}}{1-\gamma} + \left(\frac{1-\gamma^{T-N-2}}{(1-\gamma)^2} - \frac{(T-N-2)\gamma^{T-N-2}}{1-\gamma} \right) M\epsilon \\
&\leq \frac{2M}{1-\gamma}\gamma^{T-N-1} + \frac{M}{(1-\gamma)^2}\epsilon.
\end{aligned} \tag{A-21}$$

Since $M/(1-\gamma)^2$ is a constant and $\gamma^{T-N-1} \to 0$ as $T \to \infty$, $\boldsymbol{\Delta}_1$ can be arbitrarily small for a sufficiently large T.

For $\boldsymbol{\Delta}_2$. Consider

$$\left\| (\boldsymbol{U}_t - \boldsymbol{U}_\infty) \boldsymbol{V}_\infty^{T-t-1} \right\| \leq \|\boldsymbol{U}_t - \boldsymbol{U}_\infty\| \|\boldsymbol{V}_\infty\|^{T-t-1} \leq \begin{cases} \gamma^{T-t-1}\epsilon, & \text{when } t \geq N; \\ 2M\gamma^{T-t-1} & \text{when } t < N. \end{cases} \tag{A-22}$$

Therefore, when $T > N+2$, $\boldsymbol{\Delta}_2$ can be bounded as follows:

$$\begin{aligned}
\boldsymbol{\Delta}_2 &\leq \left(\sum_{t=0}^{N} + \sum_{t=N+1}^{T-1} \right) \left\| (\boldsymbol{U}_t - \boldsymbol{U}_\infty) \boldsymbol{V}_\infty^{T-t-1} \right\| \leq 2M \sum_{t=0}^{N} \gamma^{T-t-1} + \epsilon \sum_{t=N+1}^{T-1} \gamma^{T-t-1} \\
&\leq \frac{2M}{1-\gamma}\gamma^{T-N-1} + \frac{\epsilon}{1-\gamma}.
\end{aligned} \tag{A-23}$$

Since $1/(1-\gamma)$ is a constant and $\gamma^{T-N-1} \to 0$ as $T \to \infty$, $\boldsymbol{\Delta}_2$ can be arbitrarily small for a sufficiently large T.

For $\boldsymbol{\Delta}_3$. As $t \to \infty$, we have

$$\left\| \boldsymbol{U}_\infty \sum_{t=T}^{\infty} \boldsymbol{V}_\infty^t \right\| \leq \|\boldsymbol{U}_\infty\| \|\boldsymbol{V}_\infty\|^T \left\| (\boldsymbol{I} - \boldsymbol{V}_\infty)^{-1} \right\| \leq M \cdot \gamma^T \cdot \frac{1}{1-\gamma} \to 0. \tag{A-24}$$

As a result, we obtain the conclusion in Eq. (A-6). □

Theorem 3. *Suppose the loss function $\mathcal{R}$ in Eq. (3) is ℓ-smooth, lower-bounded, and has bounded gradient almost surely in the training process. Besides, assume the gradient in Eq. (4) is an*

unbiased estimator of $\nabla\mathcal{R}(\boldsymbol{\theta})$ with a bounded covariance. If the phantom gradient in Eq. (5) is an ϵ-approximation to the gradient in Eq. (4), i.e.,

$$\left\| \widehat{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right\| \leq \epsilon, \quad \textit{almost surely}, \tag{A-25}$$

then using Eq. (5) as a stochastic first-order oracle with a step size of $\eta_n = \mathcal{O}(1/\sqrt{n})$ to update $\boldsymbol{\theta}$ with gradient descent, it follows after N iterations that

$$\mathbb{E}\left[\frac{\sum_{n=1}^{N} \eta_n \|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\|^2}{\sum_{n=1}^{N} \eta_n} \right] \leq \mathcal{O}\left(\epsilon + \frac{\log N}{\sqrt{N}} \right). \tag{A-26}$$

Proof. Let $\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}}$ be the phantom gradient at the n^{th} iteration. By ℓ-smoothness of $\mathcal{R}$, we have

$$\begin{aligned} \mathcal{R}(\boldsymbol{\theta}_{n+1}) &\leq \mathcal{R}(\boldsymbol{\theta}_n) + \langle \nabla\mathcal{R}(\boldsymbol{\theta}_n), \boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n \rangle + \frac{\ell}{2} \|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n\|^2 \\ &= \mathcal{R}(\boldsymbol{\theta}_n) - \eta_n \left\langle \nabla\mathcal{R}(\boldsymbol{\theta}_n), \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\rangle + \frac{\ell \eta_n^2}{2} \left\| \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\|^2. \end{aligned} \tag{A-27}$$

Let

$$\boldsymbol{e}_n = \left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n} - \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \tag{A-28}$$

be the approximation error at the n^{th} iteration. Taking expectation *w.r.t.* the first n iterations, we have

$$\mathbb{E}_{1\sim n}\left[\mathcal{R}(\boldsymbol{\theta}_{n+1})\right] = \mathbb{E}_{1\sim n-1}\left[\mathbb{E}_n\left[\mathcal{R}(\boldsymbol{\theta}_{n+1}) \,|\, 1 \sim n-1\right]\right] = \mathbb{E}_{1\sim n-1}\left[\mathbb{E}_n\left[\mathcal{R}(\boldsymbol{\theta}_{n+1}) \,|\, \boldsymbol{\theta}_n\right]\right], \tag{A-29}$$

where the first equality comes from the *law of total expectation*, while the second from the fact that the stochasticity of the first $n-1$ steps is totally captured by the value $\boldsymbol{\theta}_n$. Consider the inner expectation in Eq. (A-29), and we omit the condition on $\boldsymbol{\theta}_n$ when no ambiguity is made. Note that in the following derivation, all expectations, variances, and covariances are conditioned on $\boldsymbol{\theta}_n$.

$$\begin{aligned} \mathbb{E}_n\left[\mathcal{R}(\boldsymbol{\theta}_{n+1})\right] &\leq \mathbb{E}_n\left[\mathcal{R}(\boldsymbol{\theta}_n) - \eta_n \left\langle \nabla\mathcal{R}(\boldsymbol{\theta}_n), \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\rangle + \frac{\ell \eta_n^2}{2} \left\| \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\|^2 \right] \\ &= \mathcal{R}(\boldsymbol{\theta}_n) - \eta_n \left\langle \nabla\mathcal{R}(\boldsymbol{\theta}_n), \mathbb{E}_n\left[\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right] \right\rangle + \frac{\ell \eta_n^2}{2} \mathbb{E}_n\left[\left\| \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\|^2 \right], \end{aligned} \tag{A-30}$$

where

$$\mathbb{E}_n\left[\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right] = \mathbb{E}_n\left[\left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n} - \boldsymbol{e}_n \right] = \nabla\mathcal{R}(\boldsymbol{\theta}_n) - \mathbb{E}_n\left[\boldsymbol{e}_n\right], \tag{A-31}$$

and

$$\mathbb{E}_n\left[\left\| \widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right\|^2 \right] = \left\| \mathbb{E}_n\left[\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right] \right\|^2 + \operatorname{tr}\left(\operatorname{Cov}_n\left(\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right) \right). \tag{A-32}$$

Suppose $\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\| \leq G$ almost surely, and then we have

$$\left\| \mathbb{E}_n\left[\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right] \right\|^2 = \|\nabla\mathcal{R}(\boldsymbol{\theta}_n) - \mathbb{E}_n\left[\boldsymbol{e}_n\right]\|^2 \leq (G+\epsilon)^2. \tag{A-33}$$

Moreover, by the properties of covariance,

$$\begin{aligned} &\operatorname{tr}\left(\operatorname{Cov}_n\left(\widehat{\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\theta}}} \right) \right) = \operatorname{tr}\left(\operatorname{Cov}_n\left(\left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n} - \boldsymbol{e}_n \right) \right) \\ &= \operatorname{tr}\left(\operatorname{Cov}_n\left(\left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n} \right) \right) + \operatorname{tr}\left(\operatorname{Cov}_n\left(\boldsymbol{e}_n \right) \right) - 2\operatorname{tr}\left(\operatorname{Cov}_n\left(\left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n}, \boldsymbol{e}_n \right) \right) \\ &\leq 2\operatorname{tr}\left(\operatorname{Cov}_n\left(\left. \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n} \right) \right) + 2\operatorname{tr}\left(\operatorname{Cov}_n\left(\boldsymbol{e}_n \right) \right), \end{aligned} \tag{A-34}$$

where the last inequility comes from

$$\begin{aligned}\left|\operatorname{tr}\left(\operatorname{Cov}\left(\boldsymbol{a},\boldsymbol{b}\right)\right)\right| &\leq \sum_i \left|\operatorname{Cov}\left(a_i,b_i\right)\right| \leq \sum_i \sqrt{\operatorname{Var}\left(a_i\right)\operatorname{Var}\left(b_i\right)} \leq \sum_i \frac{\operatorname{Var}\left(a_i\right)+\operatorname{Var}\left(b_i\right)}{2} \\ &= \frac{1}{2}\left(\operatorname{tr}\left(\operatorname{Cov}\left(\boldsymbol{a}\right)\right)+\operatorname{tr}\left(\operatorname{Cov}\left(\boldsymbol{b}\right)\right)\right).\end{aligned} \tag{A-35}$$

By the Popoviciu's inequality on variances [4], the second term in (A-34) can be bounded by $d_{\boldsymbol{\theta}}\epsilon^2$, *i.e.,*

$$\operatorname{tr}\left(\operatorname{Cov}_n\left(\boldsymbol{e}_n\right)\right) \leq d_{\boldsymbol{\theta}}\epsilon^2, \tag{A-36}$$

where $d_{\boldsymbol{\theta}}$ denotes the dimension of $\boldsymbol{\theta}$. Finally, since the gradient estimator $\partial\mathcal{L}/\partial\boldsymbol{\theta}$ has a bounded covariance, there exists $M>0$, s.t.

$$\operatorname{tr}\left(\operatorname{Cov}_n\left(\left.\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n}\right)\right) \leq M, \quad \text{almost surely.} \tag{A-37}$$

Combining (A-30), (A-31), (A-32), (A-33), (A-34), and (A-37), we have

$$\begin{aligned}\mathbb{E}_n\left[\mathcal{R}(\boldsymbol{\theta}_{n+1})\right] &\leq \mathcal{R}(\boldsymbol{\theta}_n) - \eta_n\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|^2 + \eta_n\left\langle\nabla\mathcal{R}(\boldsymbol{\theta}_n),\mathbb{E}_n\left[\boldsymbol{e}_n\right]\right\rangle + K\eta_n^2, \\ &\leq \mathcal{R}(\boldsymbol{\theta}_n) - \eta_n\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|^2 + \eta_n\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|\left\|\mathbb{E}_n\left[\boldsymbol{e}_n\right]\right\| + K\eta_n^2 \\ &\leq \mathcal{R}(\boldsymbol{\theta}_n) - \eta_n\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|^2 + \eta_n G\epsilon + K\eta_n^2,\end{aligned} \tag{A-38}$$

where $K=\ell\left(\left(G+\epsilon\right)^2+2M+2d_{\boldsymbol{\theta}}\epsilon^2\right)/2$ is a constant. Substitute (A-38) into Eq. (A-29), and it becomes

$$\mathbb{E}_{1\sim n}\left[\mathcal{R}(\boldsymbol{\theta}_{n+1})\right] \leq \mathbb{E}_{1\sim n-1}\left[\mathcal{R}(\boldsymbol{\theta}_n)\right] - \eta_n\mathbb{E}_{1\sim n-1}\left[\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|^2\right] + \eta_n G\epsilon + K\eta_n^2. \tag{A-39}$$

By taking a summation over the first N steps, we have

$$\begin{aligned}\mathbb{E}_{1\sim N}\left[\sum_{n=1}^N \eta_n\left\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\right\|^2\right] &\leq \mathcal{R}(\boldsymbol{\theta}_1) - \mathbb{E}_{1\sim N}\left[\mathcal{R}(\boldsymbol{\theta}_{N+1})\right] + G\epsilon\sum_{n=1}^N\eta_n + K\sum_{n=1}^N\eta_n^2 \\ &\leq \mathcal{R}(\boldsymbol{\theta}_1) - m + G\epsilon\sum_{n=1}^N\eta_n + K\sum_{n=1}^N\eta_n^2,\end{aligned} \tag{A-40}$$

where $m=\inf_{\boldsymbol{\theta}}\mathcal{R}(\boldsymbol{\theta})$ since $\mathcal{R}$ is lower-bounded. Dividing a factor of $\sum_{n=1}^N\eta_n$, we have

$$\mathbb{E}_{1\sim N}\left[\frac{\sum_{n=1}^N\eta_n\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\|^2}{\sum_{n=1}^N\eta_n}\right] \leq G\epsilon + \frac{\mathcal{R}(\boldsymbol{\theta}_1)-m}{\sum_{n=1}^N\eta_n} + K\frac{\sum_{n=1}^N\eta_n^2}{\sum_{n=1}^N\eta_n}. \tag{A-41}$$

Since $\eta_n=\mathcal{O}(1/\sqrt{n})$, it follows that

$$\sum_{n=1}^N\eta_n = \mathcal{O}\left(\sqrt{N}\right), \quad \frac{\sum_{n=1}^N\eta_n^2}{\sum_{n=1}^N\eta_n} = \mathcal{O}\left(\frac{\log N}{\sqrt{N}}\right). \tag{A-42}$$

Combining (A-41) and Eq. (A-42) concludes the proof. □

Remark 1. The assumption that $\mathcal{R}$ has almost-surely bounded gradient at $\{\boldsymbol{\theta}_n\}_{n=0}^N$ is reasonable. Because of the existence of norm-based regularizations, *e.g.,* weight decay, we can assume $\boldsymbol{\theta}$ is almost surely optimized within a compact set in the parameter space. If we further assume $\mathcal{R}$ is continuously differentiable, the almost-sure boundedness of $\|\nabla\mathcal{R}\|$ within the compact set follows its continuity.

Remark 2. We justify the assumption that the gradient in Eq. (4) has a bounded covariance. For the SGD algorithm, the stochasticity of the gradient in Eq. (4) comes from the random sampling of the training example (or the training mini-batch) from the dataset. Since there are finite samples in the training set, the covariance of Eq. (4) remains finite. Moreover, as Theorem 2 only considers a finite training schedule, *i.e., N* steps, the possible combination of the selected sample (or mini-batch) at each step is still finite (even though its number grows combinatorially). Therefore, it is reasonable to assume the gradient in Eq. (4) has a bounded covariance.

4 Experiment Details

In this section, we introduce the experimental settings of this paper in detail and discuss some additional findings of training implicit models.

4.1 Synthetic Setting

For the synthetic setting, the following model is used:

$$\boldsymbol{h}^* = \mathcal{F}\left(\boldsymbol{h}^* + \boldsymbol{u}\right) \tag{A-43}$$

where $\mathcal{F}$ is an 1-layer network with spectral normalization [5], and $\boldsymbol{u}$, $\boldsymbol{h}^* \in \mathbb{R}^{N\times D}$. The loss $\mathcal{L}$ is given by the mean squared error (MSE) between $\boldsymbol{h}^*$ and $\boldsymbol{y}$. We choose $N = 32$ and $D = 128$ and randomly sample 50000 data pairs $(\boldsymbol{u},\ \boldsymbol{y})$ to compute the gradient $\partial\mathcal{L}/\partial\boldsymbol{u}$.

We generate a symmetric weight matrix for the network and constrain the Lipschitz constant $L_{\boldsymbol{h}}$ to a given level using spectral normalization. For the visualization in the main paper, we adopt $L_{\boldsymbol{h}} = 0.9$. For the additional visualization on the stability of the solver in Fig. 1, we choose $L_{\boldsymbol{h}}$ from $\{0.9,\ 0.99,\ 0.999,\ 0.9999\}$.

To solve $\boldsymbol{h}^*$, we employ the fixed-point iteration as the solver. For the synthetic setting, we use 100 fixed-point iterations to obtain $\boldsymbol{h}^*$ that satisfies the relative error $\|\boldsymbol{h}^* - \mathcal{F}(\boldsymbol{h}^*, \boldsymbol{u})\|/\|\boldsymbol{h}^*\| \leq 10^{-5}$. For the visualization in Fig. 1, we also apply 100 fixed-point iterations for each $L_{\boldsymbol{h}}$.

4.2 Ablation Setting

For the ablation setting, we use the original MDEQ-Tiny [6] model (170K parameters) on CIFAR-10 [7] classification without any architecture modification. Therefore, the performance gain upon the state-of-the-art method is due to the improved training efficiency thanks to the proposed phantom gradient.

The experiments are conducted without data augmentation as in [6]. The training schedule, batch size, cosine learning rate annealing strategy, and other hyperparameters are kept unchanged for all ablation experiments. We also follow the official training protocol of MDEQ[1] to reproduce its result.

For the training protocol without pretraining, we substitute the unrolled pretraining stage by implicit differentiation. For the training protocol without Dropout, we remove the variational Dropout from the model. We also experiment with the SGD optimizer under the standard hyperparameter setting, *i.e.,* a learning rate of 0.1, a momentum of 0.9, and a weight decay of 0.0001.

We train the MDEQ model using the two types of phantom gradient with the SGD optimizer (under the hyperparameters mentioned above) and other hyperparameters unchanged from the original setting. The model is trained without shallow-layer pretraining, suggesting an $\mathcal{O}(k)$ and $\mathcal{O}(1)$ peak memory usage for the unrolling-based and the Neumann-series-based phantom gradient, respectively. In both cases, the damped fixed-point iteration starts at the solution obtained by the Broyden's method.

We monitor the Jacobian spectral radius $\rho(\partial\mathcal{F}/\partial\boldsymbol{h})$ during training for both forms of phantom gradient. It shows that the radius can grow without restriction for the state-free NPG when the phantom gradient includes high-order terms and cannot exactly match the gradient of a computational sub-graph. A similar phenomenon is observed when using the state-free gradient estimate from implicit differentiation with considerable numerical errors in the forward and backward passes [8]. On the contrary, for the state-dependent UPG, the Jacobian spectral radius is kept within a reasonable region during training thanks to the implicit Jacobian regularization.

4.3 Experiments at Scale

For large-scale experiments, we adopt MDEQ and MDEQ-Small on the CIFAR-10 [7] and ImageNet [9] benchmarks, respectively, DEQ (PostLN) [10] and DEQ (PreLN) [8] on the Wikitext-103 [11] dataset, and IGNN [12] on graph classification (COX2, PROTEINS) and node classification (PPI) benchmarks.

[1] Code available at https://github.com/locuslab/mdeq.

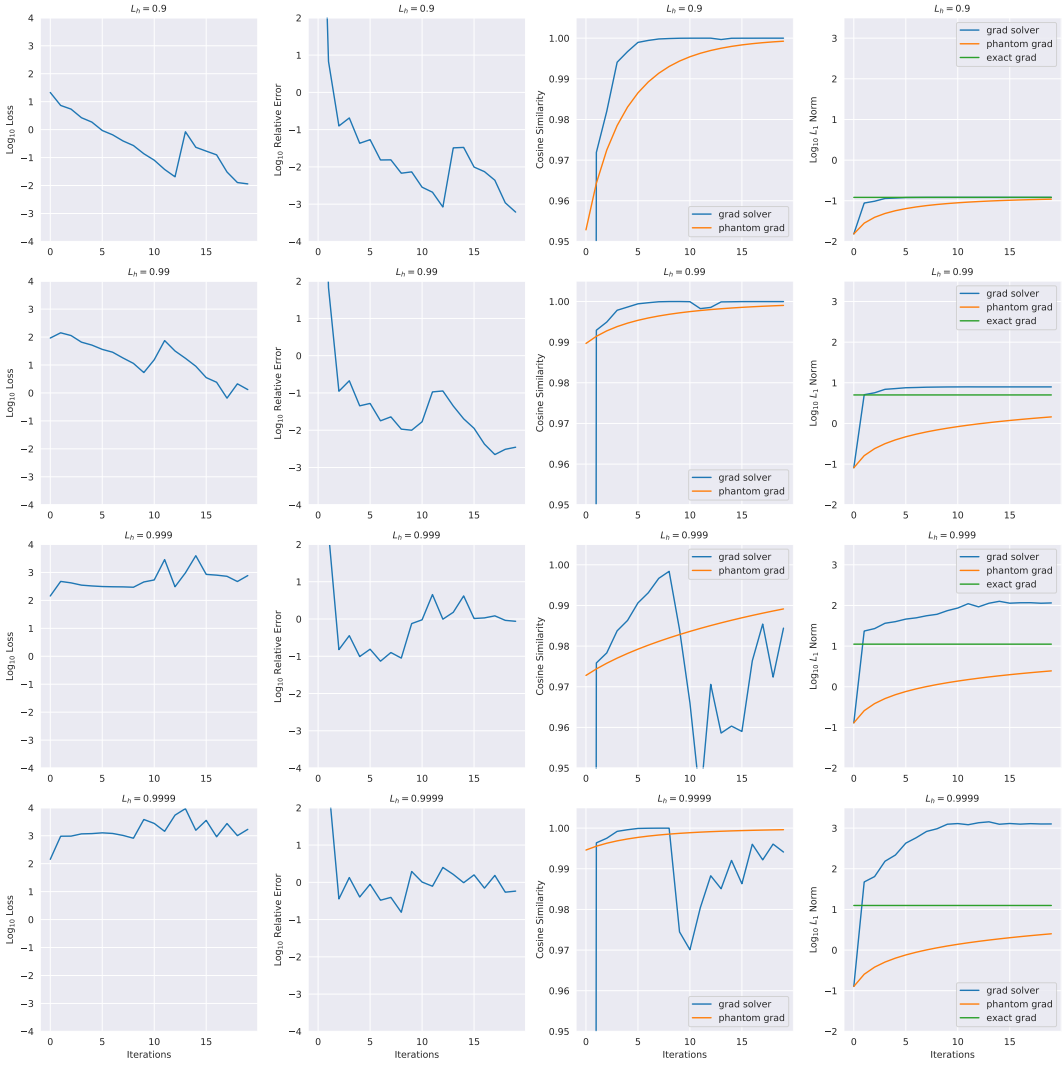

Figure 1: Visualization of gradient solvers under different $L_{\boldsymbol{h}}$.

ConvNet-based Implicit Models on Vision Datasets. To train MDEQ on CIFAR-10, we employ the UPG with $\lambda = 0.5$ and $k = 5$, *i.e.,* $\boldsymbol{A}_{5,0.5}$. Besides, we use the SGD optimizer with a learning rate of 0.1, a momentum of 0.9, and a weight decay of 0.0001, and keep other experimental settings unchanged, including the number of training epochs, the batch size, the learning rate annealing strategy, *etc.*

We adopt two settings on ImageNet. The first setting follows the practice of [6] to pretrain the model for the same number of epochs. Afterwards, the UPG with $\boldsymbol{A}_{5,0.6}$ is used to train the model for the remaining training schedule. This setting achieves a test accuracy of 75.2%. In the second setting, we adopt the UPG to train the implicit model throughout, leaving the UPG to automatically transit from the pretraining stage to the regular training stage. This setting demonstrates a test accuracy of 75.7%. The difference confirms that the automatic transition property of UPG helps alleviate the burden of hyperparameter tuning, *i.e.,* the number of steps in the pretraining stage, and benefit to the final performance as well.

We also verify the implicit Jacobian regularization from the UPG on ImageNet. By calculating the Jacobian spectral radius of the trained model on the validation set through the power method, we find that the radius $\rho(\partial \mathcal{F}_\lambda / \partial \boldsymbol{h})$ is retained around 1, although the radius of the equilibrium module $\rho(\partial \mathcal{F} / \partial \boldsymbol{h})$ usually exceeds 1 (but also remains bounded). This finding provides us with a potential path to explain why the damping operation can enhance the naive unrolling to match or even surpass

the standard implicit differentiation for ConvNets on vision tasks. We conjecture that the damping operation allows the equilibrium module to evolve within a wider range, *e.g.,* $\rho(\partial \mathcal{F}/\partial \boldsymbol{h}) > 1$, which contributes to its better representative capacity, while maintaining stability regarding the backward pass, *i.e.,* $\rho(\partial \mathcal{F}_\lambda/\partial \boldsymbol{h}) \approx 1$.

Transformer-based Implicit Models on Language Datasets. For language modeling on Wikitext-103, we follow the official training protocol of the DEQ model [10]. However, the UPG leads to inferior generalization capacity on the test set while the training loss is similar to that of implicit differentiation. The NPG even fails to optimize the DEQ (PostLN) model unless the explicit Jacobian regularization [8] or the adaptive damping factor, *e.g.,* $\lambda = 1/\rho(\partial \mathcal{F}/\partial \boldsymbol{h})$, is applied.

The performance discrepancy of different implicit models suggests the following perspective. The loss landscape and training strategy are the two sides of the same coin. Architecture, dataset, and loss function jointly define the loss landscape that has considerable impact on the preferable training strategy. For the ConvNet-based implicit model trained on vision tasks, the loss landscape is likely more regular so that the model trained on the phantom gradient can extricate itself from severe overfitting and achieve remarkable performance with acceleration despite the biased gradient estimate (which means the approximation error cannot be easily zeroed out by taking the expectation over the data distribution). For the Transformer-based implicit model on language processing tasks, in contrast, it is more arduous to employ the phantom gradient due to a lack of regularity of the loss landscape, thus inspiring us to supplement with additional regularization on the loss landscape.

To this end, we introduce the explicit Jacobian regularization (JR) [8] to strengthen the regularity of the loss landscape. The training protocol follows the official source of DEQ with JR[2]. Note that with the implicit Jacobian regularization effect of the UPG, the weight of the explicit JR can be significantly reduced, *e.g.,* from 2.0 to 0.1, and the training stability is still maintained. Meanwhile, the explicit JR can also play a vital role in alleviating overfitting for the UPG. Combining the UPG with explicit JR demonstrates an impressive test perplexity of 24.4 with $2.2\times$ training acceleration (with 14 forward Broyden iterations), and a test perplexity of 24.0 with $1.7\times$ training acceleration (with 20 forward Broyden iterations).

Our results indicate that it is more tactful to understand the training strategy combined with the loss landscape instead of only focusing on the former but neglecting the latter.

GNN-based Implicit Models on Graph Datasets. To conduct experiments on graph datasets, we follow the default architectures and training settings of the IGNN model [12][3]. We employ different damping factor λ for both graph classification and node classification. In the experiments, we encounter the training stability issue for IGNN on the PPI node classification task. Specifically, the IGNN model suffers from training collapse when using either the UPG or the exact gradient by implicit differentiation. Hence the best result from three runs is reported for this task. We conjecture that the stability issue comes from hyperparameter selection regarding the projected gradient in IGNN, since it is not easy to figure out the proper hyperparameters for well-posedness. For graph classification, the stability issue is not observed.

4.4 Additional Analysis on the Gradient Solver

To illustrate the vulnerability the gradient solver for implicit differentiation in the ill-conditioned cases, we provide the optimization dynamics in Fig. 1 and its comparison with the phantom gradient in the synthetic setting. We plot (1) the optimization objective $\|(I - \partial \mathcal{F}/\partial \boldsymbol{h})\hat{\boldsymbol{g}} - \partial \mathcal{L}/\partial \boldsymbol{h}\|$, (2) the relative error $\|(I - \partial \mathcal{F}/\partial \boldsymbol{h})\hat{\boldsymbol{g}} - \partial \mathcal{L}/\partial \boldsymbol{h}\|/\|\hat{\boldsymbol{g}}\|$, (3) the cosine similarity between the solved gradient $\hat{\boldsymbol{g}}$ (or the phantom gradient) and the exact gradient $\boldsymbol{g}$, and (4) the L_1 norm of the solved gradient $\hat{\boldsymbol{g}}$, the phantom gradient, and the exact gradient $\boldsymbol{g}$. Here, in the context of optimization, $\hat{\boldsymbol{g}}$ is the solution of the backward linear system solved by the Broyden's method.

Fig. 1 shows that the gradient solver diverges in ill-conditioned situations. It is shown that the phantom gradient demonstrates much better stability, especially in the extremely ill-conditioned cases, *e.g.,* $L_{\boldsymbol{h}} = 0.9999$. As for the Broyden's method, more optimization steps do not necessarily make the solved gradient more aligned to the exact gradient, as indicated by the oscillating cosine

[2]Code available at https://github.com/locuslab/deq.

[3]Code available at https://github.com/SwiftieH/IGNN.

similarity. Besides, the norm of the solved gradient also tends to explode in the optimization process, while the phantom gradient maintains a moderate norm throughout.

References

[1] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019. 2

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pages 265—283, 2016. 2

[3] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. math*, 3(1):133–181, 1922. 4

[4] Tiberiu Popoviciu. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9:129–145, 1935. 7

[5] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2018. 8

[6] Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Multiscale Deep Equilibrium Models. In *Neural Information Processing Systems (NeurIPS)*, pages 5238–5250, 2020. 8, 9

[7] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009. 8

[8] Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Stabilizing Equilibrium Models by Jacobian Regularization. In *International Conference on Machine Learning (ICML)*, 2021. 8, 10

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet: Large Scale Visual Recognition Challenge. *International Journal on Computer Vision (IJCV)*, 115(3):211–252, 2015. 8

[10] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep Equilibrium Models. In *Neural Information Processing Systems (NeurIPS)*, 2019. 8, 10

[11] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations (ICLR)*, 2017. 8

[12] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit Graph Neural Networks. In *Neural Information Processing Systems (NeurIPS)*, pages 11984–11995, 2020. 8, 10