MFPI-NET: LEARNING TO SOLVE ABSOLUTE VALUE EQUATIONS VIA MFPI UNROLLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Absolute value equations (AVEs) offer a unified characterization for optimality conditions across a wide range of problems, including linear and convex quadratic programming, bimatrix games, among others. Due to the non-smooth, non-convex structure and NP-hard nature induced by the absolute-value operator, solving AVEs poses significant challenges, for which many classical numerical algrithms have been proposed, and the modified fixed-point iteration (MFPI) ranks among the most effective. By combining a learnable smooth approximation of the absolute-value operator with paired neural modules regularized to preserve mutual-inverse consistency, we propose MFPI-Net, a neural solver that unrolls MFPI into learnable residual updates. Building on MFPI's convergence theory, we derive approximation error bounds for the unrolled network and establish layerwise stability against input perturbations. To understand how network depth affects the trade-off between computational cost and approximation accuracy, we analyze MFPI-Net from a continuous-time dynamical systems perspective. This motivates a two-stage framework: MFPI-Net is first leveraged to perform sign estimation to eliminate the absolute-value operator, and conventional solvers are then deployed to obtain high-accuracy solutions. Experiments demonstrate approximately $3\times$ faster high-accuracy AVE solving than MFPI, with successful solution recovery on tested instances outside the standard convergence regime. Beyond AVEs, MFPI-Net provides warm starts for various linear programming solvers through AVE reformulations.

1 INTRODUCTION

The Absolute Value Equation (AVE) is a class of linear equations involving absolute value operations (Mangasarian & Meyer, 2006). Its expressive yet simple structure supports the reformulation and solution of a wide range of problems, including the Linear Complementarity Problem (Hladík et al., 2025), Linear Programming (LP), convex quadratic programming and the Nash equilibrium for bi-matrix games (Cottle et al., 2009; Murty & Yu, 1988). Beyond its theoretical significance, AVEs have found increasing use in practical modeling, including the representation of distance distributions in traffic flow, the quantification of earnings volatility in finance, and the characterization of error-control mechanisms in engineering systems (Hladík et al., 2025; Li & Wu, 2025). Determining the solvability of an AVE is NP-hard (Mangasarian, 2007a), and deciding whether a solution is unique or multiple is NP-complete (Prokopyev, 2009). Owing to these inherent complexities, solving AVEs poses significant challenges and a variety of numerical methods have been proposed. A thorough review of classical numerical methods for solving AVEs is provided in A.1.

Deep unrolling is a technique that integrates neural networks with traditional iterative algorithms by unfolding the iterative structure into a deep network architecture. It has been widely applied in image and signal processing (Gregor & LeCun, 2010; Yang et al., 2016; Monga et al., 2021; Metwaly et al., 2024; He et al., 2023). In recent years, deep unrolling has also gained substantial attention in the field of learning to optimize, offering efficient solutions for various mathematical optimization problems. For instance, by unfolding first-order methods, (Li et al., 2024a) and (Li et al., 2024b) proposed PDHG-Net and GD-Net with theoretical guarantees to accelerate the solving of linear programming. (Xiong et al., 2025) introduced DR-GD Net, replacing linear solves in Douglas-Rachford splitting with gradient steps for efficient convex quadratic programming. A detailed review of prior work on deep unrolling is provided in A.2.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, by unrolling the Modified Fixed-Point Iteration (MFPI) method (Yu et al., 2022) — an established state-of-the-art numerical algorithm for solving AVEs — we propose MFPI-Net, a dedicated neural solver for AVE problems. We select MFPI as our base iterative scheme not only for its high computational efficiency and rigorous theoretical foundations, but more crucially, because its coupled update rules and explicit transformation operator pairs are inherently amenable to deep unrolling and neural parameterization. This favorable structural property guides our network design, allowing us to embed trainable modules directly within the original iterative framework.

Architecturally, the absolute-value operator in AVEs constitutes the main source of the problem's non-smoothness, non-convexity and NP-hardness. Hence, in designing the neural unrolling procedure, we first present a parametric smoothing strategy for the absolute-value operator. Second, inspired by the residual form of MFPI iterations, we adopt residual connections to preserve the iterative representations while introducing learnable transformations of correction terms at each layer. Notably, we substitute the critical scalar relaxation parameter in MFPI with a layerwise learnable matrix via channel unfolding. We extend the paired linear operators to two learnable nonlinear modules to increase the expressiveness of the updates, and introduce a reconstruction loss that encourages inverse consistency on sampled network states.

Theoretically, we first build upon MFPI's convergence theory to establish approximation error bounds for the unrolled network. We further derive a layerwise Lipschitz bound over bounded domains to quantify MFPI-Net's sensitivity to input-state perturbations. Beyond these standard approximation guarantees, we adopt a dynamical perspective to investigate the role of network depth. Notably, by interpreting residual updates as a discretization of continuous-time dynamics, we prove that the trajectory approximation error decreases inversely with network depth under a fixed time horizon. This motivates a two-stage framework: MFPI-Net is first leveraged to perform sign estimation to eliminate the absolute-value operator, and conventional solvers are then deployed to obtain high-accuracy solutions. When solving AVEs, this second stage reduces to solving a linear system. For other problems whose optimality conditions are equivalent to AVEs, such as linear programming, MFPI-Net can supply high-quality warm starts for the conventional second-stage solver.

We empirically evaluate MFPI-Net on AVE instances of varying dimensions as well as LP-derived AVEs. Experimental results demonstrate that MFPI-Net not only rapidly produces high-quality approximate solutions but also efficiently predicts their sign patterns. This advantage enables the proposed two-stage framework to achieve high accuracy with approximately threefold acceleration over vanilla MFPI. Moreover, MFPI-Net exhibits strong robustness: it successfully recovers solutions on tested instances lying outside the standard convergence regimes of most classical AVE solvers. For linear programming problems, the resulting neural warm starts consistently reduce both the iteration count and total solution time of a conventional LP solver across all tested problem sizes. Our main contributions are summarized as follows:

- We propose MFPI-Net, a neural architecture that unrolls MFPI for solving AVEs, integrating learnable residual updates, parametric smoothing for the absolute-value operator, and paired neural modules regularized by inverse-consistency loss.

- We provide two complementary theoretical analyses: approximation and stability bounds built upon deep-unrolling arguments, and a depth analysis based on the discretization of continuous-time dynamics.

- We develop a two-stage framework that couples MFPI-Net predictions with numerical refinement for high-accuracy solving. For AVEs, correct sign prediction reduces solution recovery to a single nonsingular linear solve; for other problems, neural predictions provide warm starts through AVE reformulations.

- We conduct extensive experiments to validate that MFPI-Net can rapidly generate high-quality approximations, enable fast, accurate two-stage solvers for AVEs and LPs, and recover solutions on AVE instances outside the standard convergence regime of classical AVE solvers.

2 MFPI-NET FOR AVE

2.1 ABSOLUTE VALUE EQUATIONS AND THE MFPI ALGORITHM

Given $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, absolute value equations take the form:

$$\mathbf{A}\mathbf{x} - |\mathbf{x}| = \mathbf{b}, \tag{1}$$

where $|\mathbf{x}| \in \mathbb{R}^n$ denoting the element-wise absolute of $\mathbf{x} \in \mathbb{R}^n$. As established in (Cottle et al., 2009; Murty & Yu, 1988), AVEs serve as a unified framework for characterizing the optimality conditions of a broad class of optimization problems. For example, the standard LP

$$\min_{\mathbf{x}} \ \mathbf{c}^\top \mathbf{x}, \qquad \text{s.t. } \mathbf{P}\mathbf{x} \geq \mathbf{h}, \ \ \mathbf{x} \geq 0, \tag{2}$$

can be transformed into the equivalent AVE:

$$(\mathbf{I} - \mathbf{M})^{-1}(\mathbf{I} + \mathbf{M})\mathbf{z} - |\mathbf{z}| = (\mathbf{M} - \mathbf{I})^{-1}\mathbf{q}, \tag{3}$$

where

$$\mathbf{M} = \begin{pmatrix} 0 & -\mathbf{P}^\top \\ \mathbf{P} & 0 \end{pmatrix} \quad \mathbf{q} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{h} \end{pmatrix}. \tag{4}$$

Similarly, the convex QP

$$\min_{\mathbf{x}} \ \mathbf{c}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x}, \qquad \text{s.t. } \mathbf{G}\mathbf{x} \geq \mathbf{h}, \ \ \mathbf{x} \geq 0, \tag{5}$$

can be converted into an AVE of the same form Eq. (3) with $\mathbf{M}$ and $\mathbf{q}$ tailored to the QP setting. See Appendix B for complete notation definitions and detailed derivations.

Given the relaxation parameter τ and the transformation matrix $\mathbf{Q}$, MFPI updates as:

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{A}^{-1}(\mathbf{Q}\mathbf{y}^{(k)} + \mathbf{b}) \\ \mathbf{y}^{(k+1)} = (1-\tau)\mathbf{y}^{(k)} + \tau\mathbf{Q}^{-1}|\mathbf{x}^{(k+1)}|, \end{cases} \quad k = 0, 1, 2, \ldots \tag{6}$$

Specifically, when $\mathbf{Q} = \mathbf{I}$, MFPI reduces to standard FPI. Its coupled two-variable updates and explicit paired operators make MFPI well suited to deep unrolling: each iteration can be mapped to a network layer, while the operator roles provide a natural structural basis for learnable neural modules. More details on FPI and MFPI are provided in Appendix C.

2.2 DESIGN OF MFPI-NET

In this subsection, we present a specific neural network architecture for MFPI unrolling, namely MFPI-Net, whose architecture is primarily composed of parameterizing the iterative process with network blocks. To enhance representation capacity, we first adopt a channel expansion technique, lifting vector iterates $\mathbf{x}^{(k)}, \mathbf{y}^{(k)} \in \mathbb{R}^{n\times 1}$ into matrix representations $\mathbf{X}^{(k)}, \mathbf{Y}^{(k)} \in \mathbb{R}^{n\times d}$. Correspondingly, the scalar parameter τ is generalized to a learnable matrix $\mathbf{W}^{(k)} \in \mathbb{R}^{d\times d}$, and the vector $\mathbf{b} \in \mathbb{R}^n$ is extended to $\mathbf{b} \cdot \mathbf{1}_{1\times d}$, where $\mathbf{1}_{1\times d} \in \mathbb{R}^{1\times d}$ denotes the all-ones vector.

Parametric smoothing of the absolute value operator. As shown in Eq. (6), the MFPI algorithm employs an alternating update scheme for $\mathbf{x}$ and $\mathbf{y}$. The inclusion of the absolute value operator $|\cdot|$ in the y-update poses challenges for seamless parameter learning in the neural network. Noting that for a real number $x \in \mathbb{R}$, $|x| = \mathrm{sgn}(x)x$, we replace the operator $|\cdot|$ with a learnable, smooth neural surrogate. Specifically, since the sign function $\mathrm{sgn}(\cdot)$ can be naturally approximated via sigmoid activations $\sigma(\cdot)$, we use a learnable surrogate φ_k at the k-th layer of network:

$$\varphi_k(x) = \sum_{l=1}^{C_\varphi} [\theta_{l,1}^k \sigma(\theta_{l,2}^k\, x + \theta_{l,3}^k) + \theta_{l,1}^k \sigma(\theta_{l,2}^k\, x - \theta_{l,3}^k) - \theta_{l,1}^k], \tag{7}$$

where C_φ is the number of subchannels, $\theta_{l,1}^k, \theta_{l,2}^k, \theta_{l,3}^k \in \mathbb{R}$ are layer-specific trainable parameters. Accordingly, $|\mathbf{X}|$ is parameterized via:

$$|\mathbf{X}| \to \varphi_k(\mathbf{X}) \odot \mathbf{X}, \tag{8}$$

where $\odot$ denotes the component-wise product.

Constrained network blocks for mutually inverse parameter matrices. The paired operators $\mathbf{Q}$ and $\mathbf{Q}^{-1}$ in MFPI provide a structural starting point for module design. We extend their respective roles in the coupled updates to learnable nonlinear mappings, denoted by $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$. These modules enrich the expressiveness of the updates without being restricted to representing a particular fixed matrix and its inverse. Both modules are implemented as multi-layer perceptrons. Taking $f_{\mathbf{Q}}$ as an example, its architecture is depicted as follows:

$$\mathbf{X}^{(l+1)} = \mathrm{PReLU}(\mathbf{X}^{(l)}\mathbf{W}_{\mathbf{Q}}^{l} + \mathbf{1}_{n\times 1}\cdot \mathbf{b}_{\mathbf{Q}}^{l}), \tag{9}$$

for $l = 0, 1, \ldots, K_{\mathbf{Q}}$. Here, $\mathbf{W}_{\mathbf{Q}}^{l}$ and $\mathbf{b}_{\mathbf{Q}}^{l}$ represent the weights and biases of the l-th layer and $K_{\mathbf{Q}}$ denotes the total number of layers. $f_{\mathbf{Q}^{-1}}$ is parameterized analogously. Inspired by the original operators' inverse relationship, we use the following global identity as an ideal consistency condition, rather than an architectural guarantee:

$$f_{\mathbf{Q}} \circ f_{\mathbf{Q}^{-1}}(\mathbf{X}) = f_{\mathbf{Q}^{-1}} \circ f_{\mathbf{Q}}(\mathbf{X}) = \mathbf{X}, \ \ \forall \mathbf{X} \in \mathbb{R}^{n\times d}. \tag{10}$$

To encourage inverse consistency on the sampled network states, we introduce a regularization term $\mathcal{L}_{\mathrm{re}}$ to quantify violations of the inverse relationship:

$$\mathcal{L}_{\mathrm{re}} = \frac{1}{2K}\sum_{k=0}^{K-1}\left[\mathrm{MSE}(\mathbf{Y}^{(k)},\ f_{\mathbf{Q}^{-1}} \circ f_{\mathbf{Q}}(\mathbf{Y}^{(k)})) + \mathrm{MSE}(\mathbf{X}^{(k)},\ f_{\mathbf{Q}} \circ f_{\mathbf{Q}^{-1}}(\mathbf{X}^{(k)}))\right], \tag{11}$$

where $\mathbf{X}^{(k)}$ and $\mathbf{Y}^{(k)}$ denote the iterative variables in the k-th layer. Minimizing $\mathcal{L}_{\mathrm{re}}$ encourages the paired modules to satisfy the inverse relationship. This regularization is further supported by our dynamical analysis in Theorem D.4, which relates inverse-consistency error to the AVE residual at equilibrium.

Architecture of MFPI-Net. Building on the channel expansion mechanism, we exploit the residual form of the MFPI iteration: each variable is updated by subtracting a correction derived from the residuals $\mathbf{A}\mathbf{x}^{(k)} - \mathbf{Q}\mathbf{y}^{(k)} - \mathbf{b}$ and $\mathbf{y}^{(k)} - \mathbf{Q}^{-1}|\mathbf{x}^{(k+1)}|$, which vanish at a fixed point satisfying $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{Q}\mathbf{y} + \mathbf{b})$ and $\mathbf{y} = \mathbf{Q}^{-1}|\mathbf{x}|$. This perspective motivates the residual connections in MFPI-Net, which retain current representations and introduce learnable transformations of the corresponding correction terms. The updated representations are then passed through PReLU activations to inject nonlinearity across layers.

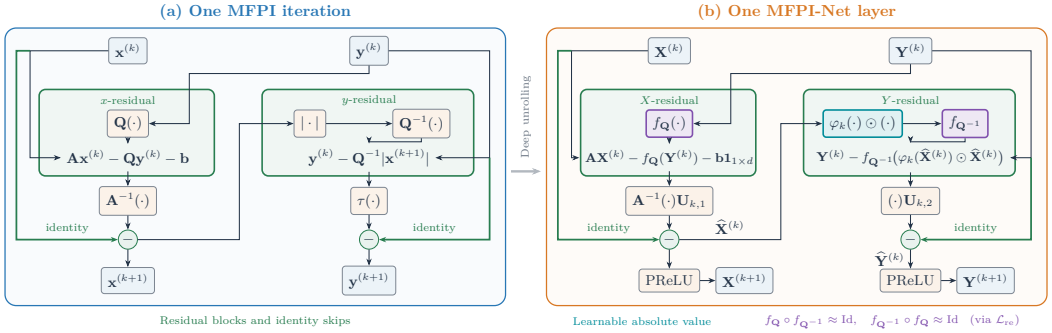


Figure 1: Correspondence between one layer of MFPI-Net and one iteration of the MFPI algorithm.

Algorithm 1 outlines the full forward procedure of MFPI-Net. Specifically, the preceding network modules are encapsulate within the residual updates in Eqs. (12,13), where $\mathbf{U}_{k,1}, \mathbf{U}_{k,2} \in \mathbb{R}^{d\times d}$ are learnable matrices that transform the correction terms along the channel dimension. Applying PReLU to $\widehat{\mathbf{X}}^{(k)}$ and $\widehat{\mathbf{Y}}^{(k)}$ yields $\mathbf{X}^{(k+1)}$ and $\mathbf{Y}^{(k+1)}$, respectively. After K iterations, a learnable linear projection $\mathbf{w} \in \mathbb{R}^{d}$ maps the expanded representation $\mathbf{X}^{(K-1)}$ back to the origin solution space via $\mathbf{x}^{(K)} = \mathbf{X}^{(K-1)}\mathbf{w}$. Figure 1 visualizes the exact layer-wise correspondence between one MFPI-Net block and a single MFPI iteration.

Algorithm 1 MFPI-Net

Require: $\mathbf{A}, \mathbf{b}$
Ensure: approximate solution $\mathbf{x}^{(K)}$
Initialize $\mathbf{X}^{(0)} = \mathbf{0}_{n\times d}$, $\mathbf{Y}^{(0)} = \mathbf{0}_{n\times d}$
for $k = 0, 1, 2, \ldots, K-2$ **do**

$$\widehat{\mathbf{X}}^{(k)} = \mathbf{X}^{(k)} - \mathbf{A}^{-1}(\mathbf{A}\mathbf{X}^{(k)} - f_Q(\mathbf{Y}^{(k)}) - \mathbf{b}\cdot\mathbf{1}_{1\times d})\mathbf{U}_{k,1} \tag{12}$$

$$\widehat{\mathbf{Y}}^{(k)} = \mathbf{Y}^{(k)} - (\mathbf{Y}^{(k)} - f_{Q^{-1}}(\varphi_k(\widehat{\mathbf{X}}^{(k)}) \odot \widehat{\mathbf{X}}^{(k)}))\mathbf{U}_{k,2} \tag{13}$$

$\mathbf{X}^{(k+1)} = \mathrm{PReLU}(\widehat{\mathbf{X}}^{(k)})$
$\mathbf{Y}^{(k+1)} = \mathrm{PReLU}(\widehat{\mathbf{Y}}^{(k)})$
end for
$\mathbf{x}^{(K)} = \mathbf{X}^{(K-1)}\mathbf{w}$
return $\mathbf{x}^{(K)} \in \mathbb{R}^n$

2.3 Network training

Given a training dataset $\mathcal{D} = \{(\mathbf{A}, \mathbf{b}, \mathbf{x}^*)\}$, where $(\mathbf{A}, \mathbf{b})$ denotes the input of an AVE instance and $\mathbf{x}^*$ is its corresponding optimal solution. For a given parameter Θ and input $(\mathbf{A}, \mathbf{b})$, we denote the K-layer output of MFPI-Net as $\mathbf{x}^{\mathrm{NN}}(\mathbf{A}, \mathbf{b}; \Theta)$. We optimize MFPI-Net end-to-end by minimizing a weighted composite objective controlled by $\lambda \in (0, 1)$, which balances mean squared error (MSE) predictions against ground truth $\mathbf{x}^*$ and incorporates the regularization term $\mathcal{L}_{\mathrm{re}}$ in Eq. (11) to enforce inverse consistency between $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$ over sampled intermediate network states. The full training loss is formulated as:

$$\min_{\Theta} \mathcal{L}_{\mathcal{D}}(\Theta) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{A},\mathbf{b},\mathbf{x}^*)\in\mathcal{D}} \big[\lambda\, \mathrm{MSE}(\mathbf{x}^{\mathrm{NN}}(\mathbf{A}, \mathbf{b}; \Theta), \mathbf{x}^*) + (1-\lambda)\mathcal{L}_{\mathrm{re}}\big]. \tag{14}$$

3 Theoretical analysis

Detailed proofs for all theoretical results in this section are provided in Appendix D.1, D.4, and D.2.

3.1 Approximation guarantees and stability of MFPI unrolling

Theorem 1 (Existence of unorlling network approximating MFPI) *Let Φ be the MFPI mapping with $\tau > 0$ and Lipschitz constant $0 < L < 1$, with fixed point $\mathbf{z}^* = (\mathbf{x}^{*\top}, (\mathbf{Q}^{-1}|\mathbf{x}^*|)^\top)^\top$ corresponding to the exact AVE solution $\mathbf{x}^*$. Fix a compact set $\mathcal{B} \subset \mathbb{R}^{2n}$ containing $\mathbf{z}^*$. For any $\epsilon > 0$ and any initial point $\mathbf{z}^{(0)}$, there exists a neural network $f_\theta : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ such that for any candidate $\mathbf{z}^*_\theta \in \mathcal{B}$ and depth $K \geq 1$:*

$$\|\mathbf{z}^* - \mathbf{z}^*_\theta\|_2 \leq \|\mathbf{R}\|_2^K \|\mathbf{z}^{(0)} - \mathbf{z}^*\|_2 + \frac{L^K}{1-L}\|\Phi(\mathbf{z}^{(0)}) - \mathbf{z}^{(0)}\|_2 + \frac{\epsilon}{1-L} + \frac{\|f_\theta(\mathbf{z}^*_\theta) - \mathbf{z}^*_\theta\|_2}{1-L}, \tag{15}$$

where $\mathbf{R} = \begin{pmatrix} 0 & \alpha\nu \\ 0 & \tau\alpha\beta\nu + |1-\tau| \end{pmatrix}$, $\nu = \|\mathbf{A}^{-1}\|_2$, $\alpha = \|\mathbf{Q}\|_2$, *and* $\beta = \|\mathbf{Q}^{-1}\|_2$.

The explicit Lipschitz continuity and contraction conditions required for the mapping Φ in the above theorem are proven in Appendix D.1. Theorem 1 guarantees the existence of a neural network with sufficient expressive power to approximate the exact MFPI iterates. Specifically, the four terms in 15 characterize distinct error sources: the first two quantify the finite-iteration MFPI error and its remaining distance to the fixed point, both of which tend to zero as depth $K \to \infty$ provided $\|\mathbf{R}\|_2 < 1$ and $L < 1$ (as ensured by Yu et al. (2022) or Lemma D.1); the third term measures the uniform neural approximation error on $\mathcal{B}$ and can be made arbitrarily small by the universal approximation theorem ; and the fourth term measures the fixed-point residual at the candidate solution, which vanishes when f_θ preserves the optimal state $f_\theta(\mathbf{z}^*) = \mathbf{z}^*$.

Theorem 2 (Arbitrary-accuracy realization of finite MFPI trajectories) *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be nonsingular, let $\mathbf{Q} = q\mathbf{I}_n$ with $q \neq 0$, and fix $\tau > 0$. For any number of MFPI steps $N \geq 1$,*

channel width $d \geq 1$, and tolerance $\epsilon > 0$, there exist finite parameters of MFPI-Net such that its states approximate all the first N iterates of Eq. (6) *to accuracy ϵ, starting from the common zero initialization.*

Theorem D.1 specializes the approximation guarantee to the MFPI-Net architecture outlined in Algorithm 1, including its PReLU activations and final channel readout. The complete statement, explicit parameter selections and proof are deferred to Appendix D.2. The construction adopts $\mathbf{Q} = q\mathbf{I}_n$, a configuration validated in the original MFPI study (Yu et al., 2022). Following standard proof paradigms for algorithm unrolling Li et al. (2024a;b), this theorem does not guarantee recoverable parameters via training, yet it rigorously proves the existence of network weights enabling arbitrary-accurate approximation of any finite MFPI trajectory.

Theorem 3 (Layerwise stability under input perturbations) *For fixed problem data with nonsingular $\mathbf{A}$ and fixed network parameters, let $\mathcal{T}_k$ denote the complete k-th update in Algorithm 1. Assume that $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$ are Lipschitz continuous with respect to the Frobenius norm. Let $\mathbf{Z} = (\mathbf{X}^\top, \mathbf{Y}^\top)^\top \in \mathbb{R}^{2n\times d}$ denote the aggregated layer state. On any set $\mathcal{S}_k \subset \mathbb{R}^{2n\times d}$ where the pre-activation state $\widehat{\mathbf{X}}^{(k)}$ is uniformly bounded, $\mathcal{T}_k$ is Lipschitz continuous: there exists a finite $\Gamma_k \geq 0$ such that, for all $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathcal{S}_k$,*

$$\|\mathcal{T}_k(\mathbf{Z}_1) - \mathcal{T}_k(\mathbf{Z}_2)\|_F \leq \Gamma_k \|\mathbf{Z}_1 - \mathbf{Z}_2\|_F. \tag{16}$$

3.2 Depth Analysis via Dynamics and Two-Stage Framework

In deep unrolling, each layer corresponds to an algorithmic iteration, so reproducing many iterations requires a deep network. Excessive network depth exacerbates training difficulty and increases inference latency, while insufficient depth often fails to meet the target solution accuracy for AVEs. To study the practical unrolling depth of MFPI-Net, we interpret its residual layer updates as discrete-time approximations of continuous dynamics (E, 2017). This perspective complements dynamical approaches for AVEs, which establish convergence to equilibria corresponding to AVE solutions under suitable conditions (Li et al., 2023). We formalize this connection for MFPI-Net by establishing equilibrium consistency and quantifying how network depth affects trajectory approximation. Let $\mathbf{B} = \mathbf{b} \cdot \mathbf{1}_{1\times d}$. Consider the coupled dynamics on a closed region $\Omega \subset \mathbb{R}^{2n\times d}$:

$$\begin{cases} \dot{\mathbf{X}} = G_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}), \\ \dot{\mathbf{Y}} = G_{\mathbf{Y}}(\mathbf{X}, \mathbf{Y}), \end{cases} \qquad \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} \in \Omega \subset \mathbb{R}^{2n\times d}, \tag{17}$$

where $G_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}) = -\mathbf{A}^{-1}(\mathbf{A}\mathbf{X} - f_{\mathbf{Q}}(\mathbf{Y}) - \mathbf{B})$, $G_{\mathbf{Y}}(\mathbf{X}, \mathbf{Y}) = -\mathbf{Y} + f_{\mathbf{Q}^{-1}}(\varphi(\mathbf{X}) \odot \mathbf{X})$.

Splitting the vector field into $F_1 = (G_{\mathbf{X}}, 0)^\top$ and $F_2 = (0, G_{\mathbf{Y}})^\top$ and applying sequential forward-Euler substeps gives

$$\begin{cases} \mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + hG_{\mathbf{X}}(\mathbf{X}^{(k)}, \mathbf{Y}^{(k)}), \\ \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + hG_{\mathbf{Y}}(\mathbf{X}^{(k+1)}, \mathbf{Y}^{(k)}). \end{cases} \tag{18}$$

By choosing suitable PReLU parameters so that the activations in both update layers act as identity maps, and setting $\mathbf{U}_{k,1} = \mathbf{U}_{k,2} = h\mathbf{I}_d$ and $\varphi_k = \varphi$, the MFPI-Net updates Eqs. (12,13) perfectly match the forward Euler discretization of Eq. (18)'s continuous dynamics. This equivalence connects discrete network layers and continuous trajectories, supporting our core theoretical bounds. Specifically, in Appendix D.4, Theorem D.3 bounds the discrepancy between the network state and the continuous trajectory at a fixed time, while Theorem D.2 controls the remaining distance from that trajectory to its equilibrium. Theorem D.4 then bounds the AVE residual at the equilibrium, which yields a bound on its distance to the exact solution when $\|\mathbf{A}^{-1}\|_2 < 1$. Combining these three estimates via the triangle inequality gives the following finite-depth solution error bound.

Theorem 4 (Combined finite-depth solution error) *Let $\mathbf{A}$ be nonsingular. Write $s(\mathbf{X}) = \varphi(\mathbf{X}) \odot \mathbf{X}$. Suppose that $f_{\mathbf{Q}}, f_{\mathbf{Q}^{-1}}$ and $s(\cdot)$ are globally Lipschitz with Lipschitz upper bounds $L_Q, L_{Q^{-1}}$ and L_s respectively, , $L_Q L_{Q^{-1}} L_s \|\mathbf{A}^{-1}\|_2 < 1$ and $\|\mathbf{A}^{-1}\|_2 < 1$. Suppose that at its unique equilibrium,*

$$\|s(\mathbf{X}_{\text{eq}}) - |\mathbf{X}_{\text{eq}}|\|_F \leq \varepsilon_1, \quad \|f_{\mathbf{Q}}(f_{\mathbf{Q}^{-1}}(s(\mathbf{X}_{\text{eq}}))) - s(\mathbf{X}_{\text{eq}})\|_F \leq \varepsilon_2, \tag{19}$$

where $\varepsilon_1, \varepsilon_2 \geq 0$. Let $\mathbf{x}^*$ be the unique solution of $\mathbf{A}\mathbf{x} - |\mathbf{x}| = \mathbf{b}$ and set $\mathbf{X}^* = \mathbf{x}^* \mathbf{1}_{1\times d}$. Fix $T > 0$ and an initial state $\mathbf{Z}_0$. There exist constants $L > 0$ and $C \geq 0$, independent of K for fixed T, modules, problem data, and initialization, such that for every integer $K \geq 1$, there is a choice of MFPI-Net parameters whose layer state $\mathbf{X}^{(K)}$, initialized at $\mathbf{Z}_0$ and evaluated before the final channel readout, satisfies

$$\|\mathbf{X}^{(K)} - \mathbf{X}^*\|_F \leq \underbrace{\frac{CT}{K}\frac{e^{LT}-1}{L}}_{\text{trajectory discretization}} + \underbrace{e^{-(1-\rho)T}(\|\mathbf{Z}_0 - \mathbf{Z}_{\text{eq}}\|_w)}_{\text{distance to equilibrium}} + \underbrace{\frac{r}{1-r}(\varepsilon_1+\varepsilon_2)}_{\text{module inconsistency}}. \tag{20}$$

An explicit parameter construction and the proof are given in Appendix D.4, Theorem D.5. Distinct from conventional unrolling error analysis, Theorem 4 rigorously characterizes the network's capacity to approximate exact AVE trajectories via dynamics. While not guaranteeing training convergence to parameter sets matching our derived error bounds, it delivers MFPI-Net's expressive power guarantee from an alternative theoretical perspective and further motivates a two-stage pipeline accelerating the overall solving process and boosting numerical efficiency.

In error bound Eq. (20), the depth K primarily controls the trajectory discretization error through the step size $h = T/K$ at fixed T. However, pseudo-transient continuation (Kelley & Keyes, 1998; Fowler & Kelley, 2005) suggests that computing an equilibrium often does not require very small time steps to closely track the transient trajectory. Instead, progressively larger pseudo-time steps can guide the iterates toward a neighborhood where local refinement is effective. Based on the above analysis, we propose the following two-stage framework. First, MFPI-Net generates a high-quality initial guess with moderate unrolling depth K:

$$\mathbf{x}^{\text{NN}} = \text{MFPI-Net}(\mathbf{A}, \mathbf{b}). \tag{21}$$

Then the second stage applies

$$\mathbf{x}^{(0)} = \mathbf{x}^{\text{NN}}, \qquad \mathbf{x}^{(k+1)} = \big(\mathbf{A} - \mathbf{D}(\mathbf{x}^{(k)})\big)^{-1}\mathbf{b}, \quad k = 0, 1, \ldots, \tag{22}$$

where $\mathbf{D}(\mathbf{x}) = \text{diag}(\text{sgn}(x_1), \ldots, \text{sgn}(x_n))$, with $\text{sgn}(0) = 0$. Moreover, if the sign pattern of the solution output by MFPI-Net in the first stage matches that of the optimal solution, i.e., $\text{sgn}(\mathbf{x}^{\text{NN}}) = \text{sgn}(\mathbf{x}^*)$, one refinement step recovers the exact solution whenever $\mathbf{A} - \mathbf{D}(\mathbf{x}^{\text{NN}})$ is nonsingular. Indeed, $\mathbf{D}(\mathbf{x}^{\text{NN}}) = \mathbf{D}(\mathbf{x}^*)$ gives

$$\big(\mathbf{A} - \mathbf{D}(\mathbf{x}^{\text{NN}})\big)\mathbf{x}^* = \mathbf{A}\mathbf{x}^* - |\mathbf{x}^*| = \mathbf{b} \quad \Rightarrow \quad \mathbf{x}^{(1)} = \big(\mathbf{A} - \mathbf{D}(\mathbf{x}^{\text{NN}})\big)^{-1}\mathbf{b} = \mathbf{x}^*.$$

4 Experimental Results

The well-known sufficient condition guaranteeing unique solvability of AVE is stated below.

Theorem 5 *(Mangasarian & Meyer, 2006) Assume that $\mathbf{A} \in \mathbb{R}^{n\times n}$ is nonsingular. If $\|\mathbf{A}^{-1}\|_2 < 1$, then the AVE* Eq. (1) *has a unique solution $\mathbf{x}^*$ for any $\mathbf{b} \in \mathbb{R}^n$.*

Classical numerical AVE algorithms are typically designed under the theoretical regime imposed by this theorem. Nevertheless, many practical AVE instances violate this assumption. In this section, we numerically validate the proposed MFPI-Net and its two-stage variant across three sets of test cases. First, we conduct standard comparisons against conventional numerical solvers on synthetic AVE instances within the above mentioned established convergence regime. The remaining two settings extend beyond this regime: one uses synthetic matrices with $\|\mathbf{A}^{-1}\|_2 \geq 1$, violating the standard unique-solvability condition; the other considers practical application scenarios by transforming LPs into AVE form, which also yields instances that do not meet the prerequisite conditions for conventional AVE solvers.

4.1 Comparison with conventional AVE solvers

We generate 10,000 AVE instances of dimension $n = 1000$ using the widely adopted construction in (Yu et al., 2022; Ke & Ma, 2017; Ke, 2020), and split them into training and validation sets in a $9:1$

Table 1: Numerical results for AVE solving with different methods.

Method		Lower-accuracy setting ($\varepsilon = 10^{-4}$) n				Higher-accuracy setting ($\varepsilon = 10^{-10}$) n			
		1000	2000	5000	10000	1000	2000	5000	10000
MFPI-Net	Time	**0.0037**	**0.0109**	**0.105**	**0.409**	**0.0048**	**0.0176**	**0.188**	**0.650**
	RES	8.11e-5	7.98e-5	7.68e-5	7.91e-5	2.05e-11	6.82e-11	5.12e-11	9.63e-11
MFPI	Time	0.0127	0.0449	0.415	2.124	0.0132	0.0491	0.439	2.221
	RES	9.11e-5	8.09e-5	9.24e-5	8.74e-5	9.33e-11	9.68e-11	9.41e-11	9.61e-11
	Iters	41.0	40.6	40.2	40.7	104.8	108.5	102.7	107.6
SORLaopt	Time	0.0123	0.0452	0.414	2.078	0.0142	0.0531	0.456	2.296
	RES	9.34e-5	9.20e-5	9.23e-5	9.36e-5	9.51e-11	9.76e-11	9.53e-11	9.81e-11
	Iters	46.1	46.0	46.0	45.6	173.8	175.8	173.3	175.6
SORLopt	Time	0.0189	0.0519	0.422	2.083	0.0208	0.0595	0.462	2.303
	RES	9.15e-5	9.04e-5	9.20e-5	9.45e-5	9.50e-11	9.72e-11	9.64e-11	9.79e-11
	Iters	46.0	45.9	45.8	45.3	172.9	175.0	172.3	174.8

ratio. We compare our method against three state-of-the-art AVE solvers based on fixed-point iteration: MFPI (Yu et al., 2022) and two SOR-type variants, SOR-Loapt (Ke & Ma, 2017) and SORLopt (Chen et al., 2024). Further details of the experimental setup are provided in Appendix E.1.

We report the performance of MFPI-Net and baseline solvers in Table 1, using the residual error (RES) between $\mathbf{A}\mathbf{x} - |\mathbf{x}|$ and $\mathbf{b}$ as the stopping criterion. The left panel shows results under the low-accuracy setting (RES $< 10^{-4}$), and the right panel shows high-accuracy results (RES $< 10^{-10}$). For each method, we report converged residual error and wall-clock solving time in seconds. For the three conventional iterative baselines, we additionally report iteration counts to convergence. All results are averaged across repeated trials, so iteration counts may be fractional. For our method, vanilla MFPI-Net is used for the low-accuracy case, and two-stage MFPI-Net for high accuracy. MFPI-Net achieves marked speedups over MFPI and other FPI-based baselines in both accuracy regimes. Notably, MFPI-Net uses only a depth of $K = 10$ (Appendix E.1), substantially fewer iterations than MFPI to attain comparable accuracy. While each unrolling layer formally corresponds to one iteration, our experiments show the required network depth can be much smaller than raw iteration counts. This highlights the promising potential of algorithm unrolling for enhancing classical iterative solvers. Appendices E.2, E.3, and E.4 provide cross-size evaluations up to $n = 25000$, component ablations, and tests under varied spectral and solution distributions; Appendix E.5 further examines refinement when the network's sign predictions contain errors.

4.2 Experiments on AVEs when $\|\mathbf{A}^{-1}\|_2 \not< 1$

In this subsection, we test our proposed method on more challenging problem instances. To this end, we retain the matrix construction used in Section 4.1 but change the singular-value sampling interval from $U(1.01, 2)$ to $U(0.8, 2.0)$. The resulting instances violate the spectral condition $\|\mathbf{A}^{-1}\|_2 < 1$, and the prescribed parameter-selection rules for the MFPI and SOR-type baselines in Section 4.1 no longer yield admissible parameters.

Table 2: Numerical results on AVE instances with $\|\mathbf{A}^{-1}\|_2 \geq 1$.

Method		n			
		1000	2000	5000	10000
MFPI-Net	Time	0.0033	0.0107	0.099	0.396
	RES	1.14e-4	1.13e-4	1.15e-4	1.13e-4
Two-stage MFPI-Net	Time	0.0038	0.0141	0.152	0.638
	RES	1.28e-9	3.99e-9	6.72e-9	1.98e-9

The update rules of standard fixed-point iteration and the generalized Newton method (Mangasarian, 2009) can still be applied to these instances. We therefore evaluate both methods, with results reported in Table 11 of Appendix E.6. Neither method converges to a solution within 10,000 iterations on the tested instances.

Table 3: LP solver performance with and without learned warm starts. Net denotes MFPI-Net; percentages in parentheses indicate reductions relative to the corresponding cold-start solver.

Size		PDLP	Net+ PDLP	cuPDLPx	Net+ cuPDLPx	HPR-LP	Net+ HPR-LP	SCS	Net+ SCS
200	Time	0.107	0.072 (↓ 32.7%)	0.036	0.020 (↓ 44.4%)	0.203	0.065 (↓ 68.0%)	0.047	0.029 (↓ 38.3%)
	Iters	611.20	**218.24** (↓ 64.3%)	104.4	56.04 (↓ 46.3%)	468.1	111.9 (↓ 76.1%)	148.5	43.5 (↓ 70.7%)
500	Time	0.677	0.380 (↓ 43.8%)	0.045	0.029 (↓ 35.6%)	0.227	0.086 (↓ 62.1%)	0.398	0.189 (↓ 52.5%)
	Iters	651.52	**184.32** (↓ 71.7%)	122.4	60.30 (↓ 50.7%)	475.0	103.8 (↓ 78.1%)	213.3	65.8 (↓ 69.2%)
1000	Time	3.266	1.611 (↓ 50.7%)	0.071	0.061 (↓ 14.1%)	0.360	0.204 (↓ 43.3%)	2.935	1.410 (↓ 52.0%)
	Iters	751.36	**179.20** (↓ 76.1%)	140.1	72.39 (↓ 48.3%)	487.1	85.7 (↓ 82.4%)	271.3	96.8 (↓ 64.3%)

In contrast, MFPI-Net successfully produces accurate approximate solutions to these challenging instances. We further use MFPI-Net outputs as a warm start for the generalized Newton method, which constitutes the two-stage strategy presented in this paper. On the tested instances, MFPI-Net correctly predicts the solution sign patterns, enabling generalized Newton to attain high-accuracy solutions in just one iteration, with residuals on the order of 10^{-9}, as reported in Table 2. These results demonstrate both the ability of MFPI-Net to solve the tested AVEs beyond the standard convergence regime and the effectiveness of the two-stage strategy. Specifically, when cold-start generalized Newton fails and vanilla MFPI cannot provide a viable warm start, MFPI-Net supplies an effective initialization that enables generalized Newton to attain high accuracy.

4.3 MFPI-Net for solving LPs

The LP-derived AVEs satisfy $\|\mathbf{A}^{-1}\|_2 = 1$ (Appendix B), outside the standard convergence regime. On the tested instances, the generalized Newton method (Mangasarian, 2009) cannot proceed because the linear system required for its update has no solution; FPI residuals remain on the order of 10^{-2} after 30,000 iterations. In contrast, MFPI-Net achieves residuals on the order of 10^{-6} in less time than FPI (Appendix E.7). These **high-quality AVE approximations provide effective LP warm starts**, which conventional solvers refine in a two-stage framework.

We evaluate PageRank-derived LPs (Nesterov, 2014) at $n = 200, 500, 1000$ using PDLP (Applegate et al., 2021), the GPU solvers cuPDLPx (Lu et al., 2025) and HPR-LP (Chen et al., 2026), and SCS (O'donoghue et al., 2016). PDLP uses a relative optimality tolerance of 10^{-4}; cuPDLPx and HPR-LP use the publicly released implementations accompanying their papers. Table 3 shows that **MFPI-Net reduces iterations and solving time for all four solvers at every tested size**, demonstrating benefits across refinement algorithms. For example, PDLP iterations decrease by 64.3%–76.1% and solving time by 32.7%–50.7%. Appendix E.8 compares against PDHG-Net warm starts, including larger instances.

For cuPDLPx and HPR-LP, relative time savings decrease with size despite substantial iteration reductions. Non-iterative overheads, including preprocessing, scaling, matrix preparation, and initialization, may limit the time savings from fewer iterations. Appendix E.9 further evaluates CQP-derived AVEs and SCS warm starts.

5 Discussion and future work

In this work, we propose MFPI-Net and its two-stage variant as efficient neural solvers for absolute value equations (AVEs), with both theoretical guarantees and extensive experiments supporting their effectiveness. We identify two promising directions for future work. First, the network construction currently relies on the inverse of $\mathbf{A}$, whose computation dominates the cost for large-scale problems; incorporating fast neural matrix inversion algorithms could substantially improve scalability. Second, although MFPI-Net empirically recovers solutions across a broad parameter regime, a rigorous convergence theory for this behavior remains to be established.

AI USE STATEMENT

We used large language models solely to polish author-written text for clarity, grammar, and readability. They were not used to develop the proposed method, formulate mathematical claims, construct proofs, generate datasets, implement algorithms, design experiments, translate text, or interpret results. The authors reviewed all AI-assisted revisions to ensure that they preserved the intended meaning and technical accuracy. We take full responsibility for the final content of this work, including all text, theoretical claims, and experimental results.

ETHICS STATEMENT

This work investigates learning-based numerical solvers for absolute value equations and their applications to optimization. The reported experiments use synthetically generated problem instances and do not involve human participants or personal data. The proposed method may benefit applications requiring repeated numerical optimization by reducing computational costs. However, its empirical performance does not guarantee reliable predictions on arbitrary problem instances, particularly outside the evaluated distributions or theoretical assumptions. Downstream applications should therefore validate numerical residuals and feasibility and employ appropriate refinement or fallback solvers when necessary.

REPRODUCIBILITY STATEMENT

We provide the architecture and forward procedure of MFPI-Net in Section 2 and Algorithm 1, together with its training objective in Section 2.3. Appendix B details the reformulations connecting optimization problems to absolute value equations, while Appendix D present the theoretical assumptions, and proofs. Appendix E.1 describes the synthetic AVE data generation procedure, network hyperparameters, evaluation metric, timing protocol, and computing environment. Appendices E.2–E.9 provide additional experimental settings and results for cross-size evaluations, component ablations, distribution variations, near-zero solution components, and optimization warm starts.

REFERENCES

Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. Modl: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, 2019.

David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34:20243–20257, 2021.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.

Louis Caccetta, Biao Qu, and Guanglu Zhou. A globally and quadratically convergent method for absolute value equations. *Computational Optimization and Applications*, 48(1):45–58, 2011.

Cairong Chen, Yinong Yang, Dongmei Yu, and Deren Han. An inverse-free dynamical system for solving the absolute value equations. *Applied Numerical Mathematics*, 168:170–181, 2021. ISSN 0168-9274. doi: https://doi.org/10.1016/j.apnum.2021.06.002.

Cairong Chen, Bo Huang, Dongmei Yu, and Deren Han. Optimal parameter of the sor-like iteration method for solving absolute value equations. *Numerical Algorithms*, 96(2):799–826, 2024.

Kaihuang Chen, Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. Hpr-lp: An implementation of an hpr method for solving linear programming: K. chen et al. *Mathematical Programming Computation*, 18(1):183–210, 2026.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Krzysztof Ciesielski. On stefan banach and some of his results. *Banach Journal of Mathematical Analysis*, 1(1):1–10, 2007.

Richard W Cottle, Jong-Shi Pang, and Richard E Stone. *The linear complementarity problem*. SIAM, 2009.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Priya Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. *International Conference on Learning Representations*, 2021.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017. doi: 10.1007/s40304-017-0103-z.

K. R. Fowler and C. T. Kelley. Pseudo-transient continuation for nonsmooth nonlinear equations. *SIAM Journal on Numerical Analysis*, 43(4):1385–1406, 2005. doi: 10.1137/S0036142903431298.

Xi Gao, Jinxin Xiong, Akang Wang, Qihong Duan, Jiang Xue, and Qingjiang Shi. Ipm-lstm: A learning-based interior point method for solving nonlinear programs. *Advances in Neural Information Processing Systems*, 37:122891–122916, 2024.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406, 2010.

Peng Guo, Shi-Liang Wu, and Cui-Xia Li. On the sor-like iteration method for solving absolute value equations. *Applied mathematics letters*, 97:107–113, 2019.

Zhao He, Ya-Nan Zhu, Yu Chen, Yi Chen, Yuchen He, Yuhao Sun, Tao Wang, Chengcheng Zhang, Bomin Sun, Fuhua Yan, et al. A deep unrolled neural network for real-time mri-guided brain intervention. *Nature Communications*, 14(1):8257, 2023.

Milan Hladík, Hossein Moosaei, Fakhrodin Hashemi, Saeed Ketabchi, and Panos M Pardalos. An overview of absolute value equations: from theory to solution methods and challenges. *Computational Optimization and Applications*, pp. 1–54, 2025.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Xingxing Ju, Chuandong Li, Xin Han, and Xing He. Neurodynamic network for absolute value equations: A fixed-time convergence technique. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(3):1807–1811, 2022. doi: 10.1109/TCSII.2021.3128416.

Yi-Fen Ke and Chang-Feng Ma. Sor-like iteration method for solving absolute value equations. *Applied Mathematics and Computation*, 311:195–202, 2017.

Yifen Ke. The new iteration algorithm for absolute value equation. *Applied Mathematics Letters*, 99:105990, 2020.

C. T. Kelley and David E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998. doi: 10.1137/S0036142996304796.

Bingheng Li, Linxin Yang, Yupeng Chen, Senmiao Wang, Haitao Mao, Qian Chen, Yao Ma, Akang Wang, Tian Ding, Jiliang Tang, et al. Pdhg-unrolled learning-to-optimize method for large-scale linear programming. *Proceedings of the 41st International Conference on Machine Learning*, pp. 29164–29180, 2024a.

Cui-Xia Li and Shi-Liang Wu. An application of generalized absolute value equation. *Journal of Nonlinear Mathematical Physics*, 32(1):1–23, 2025.

Qian Li, Tian Ding, Linxin Yang, Minghui Ouyang, Qingjiang Shi, and Ruoyu Sun. On the power of small-size graph neural networks for linear programming. *the 38-th Annual Conference on Neural Information Processing Systems*, 2024b.

Xuehua Li, Dongmei Yu, Yinong Yang, Deren Han, and Cairong Chen. A fixed-time inverse-free dynamical system for solving the system of absolute value equations. *Numerical Mathematics: Theory, Methods and Applications*, 2023. doi: 10.4208/nmtma.OA-2022-0148.

Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. *International Conference on Learning Representations (ICLR)*, 2019.

Haihao Lu, Zedong Peng, and Jinwen Yang. cupdlpx: A further enhanced gpu-based first-order solver for linear programming. *arXiv preprint arXiv:2507.14051*, 2025.

Olvi L Mangasarian. Absolute value programming. *Computational Optimization and Applications*, 36:43–53, 2007a.

Olvi L Mangasarian. Absolute value equation solution via concave minimization. *Optimization Letters*, 1(1):3–8, 2007b.

Olvi L Mangasarian. A generalized newton method for absolute value equations. *Optimization Letters*, 3(1):101–108, 2009.

Olvi L Mangasarian. Primal-dual bilinear programming solution of the absolute value equation. *Optimization Letters*, 6(7):1527–1533, 2012.

Olvi L Mangasarian. Absolute value equation solution via dual complementarity. *Optimization Letters*, 7(4):625–630, 2013.

Olvi L Mangasarian. A hybrid algorithm for solving the absolute value equation. *Optimization Letters*, 9(7):1469–1474, 2015.

Olvi L Mangasarian and RR Meyer. Absolute value equations. *Linear Algebra and Its Applications*, 419(2-3):359–367, 2006.

Kareem Metwaly, Junho Kweon, Khaled Alhujaili, Fulvio Gini, Maria Sabrina Greco, Muralidhar Rangaswamy, and Vishal Monga. Mimo radar beampattern design via algorithm unrolling. *IEEE Transactions on Aerospace and Electronic Systems*, 60(6):9204–9220, 2024.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

Katta G Murty and Feng-Tien Yu. *Linear complementarity, linear and nonlinear programming*, volume 3. Heldermann Berlin, 1988.

Yu Nesterov. Subgradient methods for huge-scale optimization problems. *Mathematical Programming*, 146(1):275–297, 2014.

Brendan O'donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

Oleg Prokopyev. On equivalent reformulations for absolute value equations. *Computational Optimization and Applications*, 44:363–372, 2009.

Jiri Rohn, Vahideh Hooshyarbakhsh, and Raena Farhadsefat. An iterative method for solving absolute value equations and sufficient conditions for unique solvability. *Optimization Letters*, 8(1): 35–44, 2014.

Jinxin Xiong, Xi Gao, Linxin Yang, Jiang Xue, Xiaodong Luo, and Akang Wang. Solving quadratic programs via deep unrolled douglas-rachford splitting. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Linxin Yang, Bingheng Li, Tian Ding, Jianghua Wu, Akang Wang, Yuyi Wang, Jiliang Tang, Ruoyu Sun, and Xiaodong Luo. An efficient unsupervised framework for convex quadratic programs via deep unrolling. *arXiv preprint arXiv:2412.01051*, 2024.

Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep {ADMM-{Net for compressive sensing mri. *Advances in Neural Information Processing Systems*, 29, 2016.

Dongmei Yu, Cairong Chen, and Deren Han. A modified fixed point iteration method for solving the system of absolute value equations. *Optimization*, 71(3):449–461, 2022.

Dongmei Yu, Gehao Zhang, Cairong Chen, and Deren Han. The neural network models with delays for solving absolute value equations. *Neurocomputing*, 589:127707, 2024.

Moslem Zamani and Milan Hladík. A new concave minimization algorithm for the absolute value equation solution. *Optimization Letters*, 15(6):2241–2254, 2021.

A Related work

A.1 Numerical methods for solving AVE

The numerical solution of absolute value equations (AVEs) originated from a series of foundational studies by Mangasarian and his collaborators. Building on the theoretical foundation established by Mangasarian & Meyer (2006), Mangasarian developed several pioneering reformulations and solution methods, including a concave minimization formulation with a successive linear programming method (Mangasarian, 2007b), a nonsmooth root-finding formulation addressed by a generalized Newton method (Mangasarian, 2009), and subsequent approaches based on primal-dual bilinear programming (Mangasarian, 2012), dual complementarity and (Mangasarian, 2013), and hybrid schemes (Mangasarian, 2015).

Following Mangasarian's foundational work, a rich body of methods for solving AVEs has been developed. Caccetta et al. (2011) developed a quadratically convergent smoothing Newton method for systems with singular values exceeding one. Rohn et al. (2014) leveraged matrix splitting techniques to introduce Picard and Picard-HSS iterative schemes, respectively. Building on this line of work, Ke & Ma (2017) proposed SOR-like iterations to accelerate convergence, a framework for which Guo et al. (2019) later derived new convergence conditions. In more recent years, Ke (2020) reformulated the AVE into a two-by-two block nonlinear system and proposed an FPI scheme, which Yu et al. (2022) subsequently improved by introducing MFPI that incorporates parameter matrices to widen the convergence range. Zamani & Hladík (2021) refined Mangasarian's concave minimization framework to enhance its convergence behavior across a broader range of AVE cases. Most recently, Chen et al. (2024) revisited the SOR-like iterations, delivering a refined convergence analysis and deriving iteration-independent optimal parameters. We note that quite a few of the aforementioned approaches belong to the family of fixed-point iterative algorithms. Beyond FPI and its variants, Picard-type iteration and SOR-like algorithms also fall into this category.

Parallel to discrete numerical iterations, continuous-time dynamical systems and neurodynamic networks have emerged for solving AVEs. Chen et al. (2021) proposed an inverse-free, globally asymptotically stable dynamical system with projection-type error bounds. Ju et al. (2022) introduced a projection neurodynamic network that achieves fixed-time convergence under mild conditions. Yu et al. (2024) developed an inverse-free neural network with mixed delays, establishing its exponential stability and strong performance on complex AVE instances. Meanwhile, algorithm unrolling has emerged as a technique that integrates neural networks with traditional iterative algorithms by unfolding iterative structures into structured network architectures, and it has achieved remarkable performance on several problem-solving tasks. Among the rich set of conventional AVE solvers, selecting a suitable algorithm for unrolling would be a novel route to building AVE neural solvers, one that can preserve the interpretability of classical iterative methods while harnessing the expressive power of deep learning. Therefore, in this work we investigate how to unroll fixed-point iterative algorithms into networks for efficient AVE solving, and leverage dynamical systems perspectives to enrich the analysis of deep unrolling.

A.2 Deep Unrolling

Deep unrolling is an emerging deep learning technique that incorporates neural networks with traditional iterative algorithms. By designing network architectures based on conventional iterative structures, deep unrolling leverages the strengths of both approaches, improving the performance of neural networks while enhancing the computational efficiency of classical algorithms. This technique was pioneered by Gregor & LeCun (2010) for solving LASSO regression. By unrolling the

ISTA algorithm, they introduced the LISTA (Learned-ISTA), which substantially improved computational speed. Subsequently, Yang et al. (2016) unfolded the ADMM method for CS-MRI into a deep network architecture, termed ADMM-Net, to tackle the limitations of conventional CS-MRI methods, such as reliance on manual parameter tuning, difficulty in reconstructing complex images, and the trade-off between reconstruction accuracy and processing speed. Currently, a series of deep unfolded algorithms, exemplified by LISTA and ADMM-Net, have played an important role in signal and image processing Aggarwal et al. (2019); Metwaly et al. (2024); He et al. (2023); Liu & Chen (2019).

In recent years, Learning to Optimize has emerged as a prominent research direction Chen et al. (2022). The interpretability of neural networks derived from Deep unrolling facilitates the analysis of their theoretical properties; thus, many studies have sought to apply Deep unrolling to the solution of mathematical optimization problems. For instance, building upon the unrolling of first-order methods for linear programming, Li et al. (2024a) developed PDHG-Net, which offers strong theoretical guarantees and improves the efficiency of solving large-scale linear programming problems. Li et al. (2024b) proposed the GD-Net, which is theoretically capable of solving LPs with polylogarithmic-depth and constant-width. Based on the PDQP algorithm, Yang et al. (2024) designed PDQP-Net for convex quadratic programming. Their analysis demonstrates that PDQP-Net inherits the linear convergence of the PDQP algorithm while approximating the optimal solution with a logarithmic dependence on the number of neurons. Gao et al. (2024) utilized deep unrolling to transform the primal-dual interior point method (IPM) into a neural network. By leveraging an LSTM to approximate linear system solutions within the IPM, they proposed IPM-LSTM for solving non-linear programmings (such as QPs and QCQPs). Their approach warm-starts IPOPT, reducing iterations by up to 60% and total solution time by up to 70%.

B Transform Optimization Models into AVE for Solution

According to (Cottle et al., 2009; Murty & Yu, 1988), problems such as linear programming, convex quadratic programming and Nash equilibrium for bi-matrix games can be transformed into linear complementarity problems (LCPs). Under the invertibility conditions specified below, these LCPs can subsequently be transformed into AVEs; for bimatrix games, the trivial LCP solution must be excluded.

B.1 Transform Linear Programming into AVE for Solution

We consider the following Linear Programming (LP) problem in canonical form:

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{P}\mathbf{x} \geq \mathbf{h}, \\ & \mathbf{x} \geq 0, \end{aligned} \tag{B.1}$$

where $\mathbf{P} \in \mathbb{R}^{m\times n}$, $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{h} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$. According to the theorem of complementary slackness (Bertsimas & Tsitsiklis, 1997), $\mathbf{x}$ is a primal optimal solution if and only if there exists $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{x}, \mathbf{y} \geq 0$ and the dual and primal slack vectors $\mathbf{u} = \mathbf{c} - \mathbf{P}^\top \mathbf{y} \geq 0$ and $\mathbf{v} = \mathbf{P}\mathbf{x} - \mathbf{h} \geq 0$ satisfy

$$\begin{aligned} \mathbf{u}_i \mathbf{x}_i = (\mathbf{c}_i - (\mathbf{P}^\top \mathbf{y})_i)\mathbf{x}_i = 0, \quad i = 1, 2, \ldots, n, \\ \mathbf{v}_j \mathbf{y}_j = ((\mathbf{P}\mathbf{x})_j - \mathbf{h}_j)\mathbf{y}_j = 0, \quad j = 1, 2, \ldots, m. \end{aligned} \tag{B.2}$$

Here subscripts denote vector components. These conditions can be reformulated as follows:

$$\begin{aligned} &\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} - \begin{pmatrix} 0 & -\mathbf{P}^\top \\ \mathbf{P} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{h} \end{pmatrix} \\ &\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \geq 0 \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \geq 0 \quad (\mathbf{u}^\top, \mathbf{v}^\top) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = 0, \end{aligned} \tag{B.3}$$

which constitutes a standard-form linear complementarity problem (LCP):

$$\begin{aligned} &\mathbf{w}_1 - \mathbf{M}\mathbf{w}_2 = \mathbf{q}, \quad \mathbf{w}_1, \mathbf{w}_2 \geq 0, \quad \mathbf{w}_1^\top \mathbf{w}_2 = 0 \\ &\mathbf{w}_1 = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}, \ \mathbf{w}_2 = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \ \mathbf{M} = \begin{pmatrix} 0 & -\mathbf{P}^\top \\ \mathbf{P} & 0 \end{pmatrix}, \ \mathbf{q} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{h} \end{pmatrix}. \end{aligned} \tag{B.4}$$

Leveraging the equivalence between AVE and LCP (Cottle et al., 2009), we define $\mathbf{w}_1 = |\mathbf{z}| - \mathbf{z}, \quad \mathbf{w}_2 = \mathbf{z} + |\mathbf{z}|$. These vectors are nonnegative and complementary; conversely, any complementary nonnegative pair has this form with $\mathbf{z} = (\mathbf{w}_2 - \mathbf{w}_1)/2$. Substitution gives $(\mathbf{I} - \mathbf{M})|\mathbf{z}| - (\mathbf{I} + \mathbf{M})\mathbf{z} = \mathbf{q}$. Since the LP matrix $\mathbf{M}$ is skew-symmetric, $\mathbf{I} - \mathbf{M}$ is nonsingular, and the LCP is transformed into an AVE of the form:

$$(\mathbf{I} - \mathbf{M})^{-1}(\mathbf{I} + \mathbf{M})\mathbf{z} - |\mathbf{z}| = (\mathbf{M} - \mathbf{I})^{-1}\mathbf{q}, \tag{B.5}$$

where $\mathbf{I} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the identity matrix. Conversely, if $\mathbf{z}^*$ is a solution of this AVE, then $\mathbf{w}_1^* = |\mathbf{z}^*| - \mathbf{z}^*, \mathbf{w}_2^* = \mathbf{z}^* + |\mathbf{z}^*|$ solve the LCP in Eq.(B.4), and $\mathbf{w}_2^*$ provides the concatenation of the primal and dual optimal solutions for the LP problem in Eq.(B.1). AVE instances obtained by this LP reformulation fail to satisfy the sufficient condition $\|\mathbf{A}^{-1}\|_2 < 1$; this does not imply that a particular instance has multiple solutions.

Theorem B.1 *For any AVE instance obtained from an LP by the reformulation above,* $\|\mathbf{A}^{-1}\|_2 = 1$. *Hence it fails to satisfy the sufficient condition for unique solvability* $\|\mathbf{A}^{-1}\|_2 < 1$.

Proof. Since $\mathbf{M}^\top = -\mathbf{M}$, both $\mathbf{I} - \mathbf{M}$ and $\mathbf{I} + \mathbf{M}$ are nonsingular. These matrices and their inverses commute, so for $\mathbf{A} = (\mathbf{I} - \mathbf{M})^{-1}(\mathbf{I} + \mathbf{M})$ we have $\mathbf{A}^\top = (\mathbf{I} - \mathbf{M})(\mathbf{I} + \mathbf{M})^{-1} = \mathbf{A}^{-1}$. Thus $\mathbf{A}$ is orthogonal and $\|\mathbf{A}^{-1}\|_2 = 1$. □

B.2 Transforming Convex Quadratic Programming into AVE for Solution

Consider the quadratic programming in the following form:

$$\begin{aligned} \min \quad & \mathbf{c}^\top\mathbf{x} + \frac{1}{2}\mathbf{x}^\top\mathbf{P}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{G}\mathbf{x} \geq \mathbf{h}, \\ & \mathbf{x} \geq 0, \end{aligned} \tag{B.6}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{G} \in \mathbb{R}^{m\times n}$, and $\mathbf{h} \in \mathbb{R}^m$. When the matrix $\mathbf{P}$ is symmetric positive semidefinite (PSD), the problem in Eq.(B.6) is a convex quadratic program. By introducing $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{u} \in \mathbb{R}^n$ as Lagrange multipliers, we derive the corresponding Lagrangian function:

$$\mathbf{c}^\top\mathbf{x} + \frac{1}{2}\mathbf{x}^\top\mathbf{P}\mathbf{x} - \mathbf{y}^\top(\mathbf{G}\mathbf{x} - \mathbf{h}) - \mathbf{u}^\top\mathbf{x} \tag{B.7}$$

For the convex quadratic program in Eq.(B.6), $\mathbf{x} \in \mathbb{R}^n$ is an optimal solution if and only if there exist multipliers $\mathbf{u}, \mathbf{y}$ satisfying the following Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} \text{primal feasibility}: \quad & \mathbf{G}\mathbf{x} - \mathbf{h} \geq 0, \quad \mathbf{x} \geq 0 \\ \text{dual feasibility}: \quad & \mathbf{u} \geq 0, \quad \mathbf{y} \geq 0 \\ \text{complementary slackness}: \quad & \mathbf{y}^\top(\mathbf{G}\mathbf{x} - \mathbf{h}) = 0, \quad \mathbf{u}^\top\mathbf{x} = 0 \\ \text{stationarity}: \quad & \mathbf{c} + \mathbf{P}\mathbf{x} - \mathbf{G}^\top\mathbf{y} - \mathbf{u} = 0. \end{aligned} \tag{B.8}$$

By defining $\mathbf{v} = \mathbf{G}\mathbf{x} - \mathbf{h}$, the Karush-Kuhn-Tucker (KKT) conditions are equivalent to:

$$\begin{aligned} & \mathbf{v} \geq 0, \quad \mathbf{x} \geq 0 \\ & \mathbf{u} \geq 0, \quad \mathbf{y} \geq 0 \\ & \mathbf{y}^\top\mathbf{v} = 0, \quad \mathbf{u}^\top\mathbf{x} = 0 \\ & \mathbf{c} + \mathbf{P}\mathbf{x} - \mathbf{G}^\top\mathbf{y} - \mathbf{u} = 0. \end{aligned} \tag{B.9}$$

Furthermore, this system can be reformulated as:

$$\begin{aligned} & \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} - \begin{pmatrix} \mathbf{P} & -\mathbf{G}^\top \\ \mathbf{G} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{h} \end{pmatrix} \\ & \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \geq 0 \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \geq 0 \quad (\mathbf{u}^\top,\ \mathbf{v}^\top) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = 0, \end{aligned} \tag{B.10}$$

which constitutes a standard-form LCP:

$$\begin{gathered}\mathbf{w}_1 - \mathbf{M}\mathbf{w}_2 = \mathbf{q}, \quad \mathbf{w}_1, \mathbf{w}_2 \geq 0, \quad \mathbf{w}_1^\top \mathbf{w}_2 = 0 \\ \mathbf{w}_1 = \begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix},\ \mathbf{w}_2 = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix},\ \mathbf{M} = \begin{pmatrix}\mathbf{P} & -\mathbf{G}^\top\\ \mathbf{G} & \mathbf{0}\end{pmatrix},\ \mathbf{q} = \begin{pmatrix}\mathbf{c}\\ -\mathbf{h}\end{pmatrix}.\end{gathered} \tag{B.11}$$

Provided that $\mathbf{I} - \mathbf{M}$ is nonsingular, the LCP can be transformed into the AVE in Eq. (B.5). This invertibility is not guaranteed by convexity alone. If necessary, replace $(\mathbf{M}, \mathbf{q}, \mathbf{w}_1)$ by $(\eta\mathbf{M}, \eta\mathbf{q}, \eta\mathbf{w}_1)$ with $\eta > 0$ sufficiently small that $\eta\|\mathbf{M}\|_2 < 1$. This preserves the LCP solutions in $\mathbf{w}_2$ and ensures invertibility of $\mathbf{I} - \eta\mathbf{M}$. Consequently, convex quadratic programming can also be addressed via the AVE framework. However, AVE instances derived from quadratic programming probably fail to satisfy $\|\mathbf{A}^{-1}\|_2 < 1$. To illustrate this, consider the following quadratic program as an example:

$$\begin{aligned}\min \quad & \frac{3}{2}x_1^2 + \frac{3}{2}x_2^2 \\ \text{s.t.} \quad & 3x_1 \geq 1 \\ & x_1, x_2 \geq 0.\end{aligned} \tag{B.12}$$

In this instance, the matrices are $\mathbf{P} = \begin{pmatrix}3 & 0\\ 0 & 3\end{pmatrix}$ and $\mathbf{G} = \begin{pmatrix}3 & 0\end{pmatrix}$. The resulting AVE derived from this quadratic program takes the form:

$$\mathbf{A} = (\mathbf{I} - \mathbf{M})^{-1}(\mathbf{I} + \mathbf{M}), \qquad \mathbf{M} = \begin{pmatrix}\mathbf{P} & -\mathbf{G}^\top\\ \mathbf{G} & \mathbf{0}\end{pmatrix} = \begin{pmatrix}3 & 0 & -3\\ 0 & 3 & 0\\ 3 & 0 & 0\end{pmatrix}. \tag{B.13}$$

from which we obtain

$$\mathbf{A}^{-1} = (\mathbf{I} + \mathbf{M})^{-1}(\mathbf{I} - \mathbf{M}) = \begin{pmatrix}-\frac{11}{13} & 0 & \frac{6}{13}\\ 0 & -\frac{1}{2} & 0\\ -\frac{6}{13} & 0 & -\frac{5}{13}\end{pmatrix}. \tag{B.14}$$

In this case, $\|\mathbf{A}^{-1}\|_2 = 1$.

B.3 Transforming Bimatrix Games into AVE for Nash Equilibrium

Consider the bimatrix game $\Gamma(\mathbf{U}, \mathbf{V})$, where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{m\times n}$ denote the payoff matrices for the two players, respectively. We assume that both payoff matrices are entrywise strictly positive; this can always be achieved by adding a constant to each player's payoffs without changing the Nash equilibria. A strategy profile $(\mathbf{s}^*, \mathbf{t}^*)$, with $\mathbf{s}^*, \mathbf{t}^* \geq 0$ and $\mathbf{1}_m^\top \mathbf{s}^* = \mathbf{1}_n^\top \mathbf{t}^* = 1$, is a Nash equilibrium if and only if

$$\begin{aligned}&(\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^* \geq \mathbf{s}^\top U\mathbf{t}^* \qquad \text{for all } \mathbf{s} \geq 0 \quad \text{and} \quad \sum_{i=1}^m \mathbf{s}_i = 1 \\ &(\mathbf{s}^*)^\top \mathbf{V}\mathbf{t}^* \geq (\mathbf{s}^*)^\top \mathbf{V}\mathbf{t} \qquad \text{for all } \mathbf{t} \geq 0 \quad \text{and} \quad \sum_{j=1}^n \mathbf{t}_j = 1.\end{aligned} \tag{B.15}$$

Now, consider a LCP of the following form:

$$\begin{aligned}&\mathbf{u} = \mathbf{1}_m - \mathbf{U}\mathbf{t} \geq 0, \quad \mathbf{s} \geq 0, \quad \mathbf{s}^\top \mathbf{u} = 0 \\ &\mathbf{v} = \mathbf{1}_n - \mathbf{V}^\top \mathbf{s} \geq 0, \quad \mathbf{t} \geq 0, \quad \mathbf{t}^\top \mathbf{v} = 0,\end{aligned} \tag{B.16}$$

where $\mathbf{1}_m$ and $\mathbf{1}_n$ denote the vectors of all-ones in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

If $(\mathbf{s}^*, \mathbf{t}^*)$ is a Nash equilibrium strategy profile, consider

$$\mathbf{s}' = \mathbf{s}^*/(\mathbf{s}^*)^\top \mathbf{V}\mathbf{t}^* \quad \text{and} \quad \mathbf{t}' = \mathbf{t}^*/(\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^*. \tag{B.17}$$

For any $\mathbf{s} \in \mathbb{R}^m$ and $\mathbf{t} \in \mathbb{R}^n$ such that $\mathbf{s}, \mathbf{t} \geq 0$, $\sum_{i=1}^m \mathbf{s}_i = 1$ and $\sum_{j=1}^n \mathbf{t}_j = 1$, it follows that

$$\begin{aligned}&\mathbf{s}^\top(\mathbf{1}_m - \mathbf{U}\mathbf{t}') = 1 - \mathbf{s}^\top \mathbf{U}\mathbf{t}^*/(\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^* \geq 0, \\ &\mathbf{t}^\top(\mathbf{1}_n - \mathbf{V}^\top \mathbf{s}') = 1 - \mathbf{t}^\top \mathbf{V}^\top \mathbf{s}^*/(\mathbf{s}^*)^\top \mathbf{V}\mathbf{t}^* \geq 0.\end{aligned} \tag{B.18}$$

This indicates $\mathbf{u} = \mathbf{1}_m - \mathbf{U}\mathbf{t}' \geq 0$ and $\mathbf{v} = \mathbf{1}_n - \mathbf{V}^\top \mathbf{s}' \geq 0$. Besides, we have

$$\begin{aligned}(\mathbf{s}')^\top \mathbf{u} &= \frac{1 - (\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^* / ((\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^*)}{(\mathbf{s}^*)^\top \mathbf{V}\mathbf{t}^*} = 0, \\ (\mathbf{t}')^\top \mathbf{v} &= \frac{1 - (\mathbf{t}^*)^\top \mathbf{V}^\top \mathbf{s}^* / ((\mathbf{s}^*)^\top \mathbf{V}\mathbf{t}^*)}{(\mathbf{s}^*)^\top \mathbf{U}\mathbf{t}^*} = 0.\end{aligned} \tag{B.19}$$

Thus, $(\mathbf{s}', \mathbf{t}')$ is the solution to LCP in Eq. (B.16). Conversely, let $(\mathbf{s}', \mathbf{t}') \neq (0, 0)$ be a solution of the LCP in Eq. (B.16). If either vector were zero, complementarity would force the other to be zero, so both have strictly positive component sums. The corresponding Nash equilibrium strategy profile is recovered via:

$$\mathbf{s}^* = \mathbf{s}' / \mathbf{1}_m^\top \mathbf{s}' \quad \text{and} \quad \mathbf{t}^* = \mathbf{t}' / \mathbf{1}_n^\top \mathbf{t}'. \tag{B.20}$$

Indeed, writing $a = \mathbf{1}_m^\top \mathbf{s}'$ and $b = \mathbf{1}_n^\top \mathbf{t}'$, feasibility gives $\mathbf{U}\mathbf{t}^* \leq \mathbf{1}_m / b$ and $\mathbf{V}^\top \mathbf{s}^* \leq \mathbf{1}_n / a$, with equality on the supports of $\mathbf{s}^*$ and $\mathbf{t}^*$ by complementarity. Thus both players are best responding. Searching for a Nash equilibrium is therefore equivalent to finding a nonzero solution of the following standard LCP:

$$\begin{gathered}\mathbf{w}_1 - \mathbf{M}\mathbf{w}_2 = \mathbf{q}, \quad \mathbf{w}_1, \mathbf{w}_2 \geq 0, \quad \mathbf{w}_1^\top \mathbf{w}_2 = 0 \\ \mathbf{w}_1 = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}, \ \mathbf{w}_2 = \begin{pmatrix} \mathbf{s} \\ \mathbf{t} \end{pmatrix}, \ \mathbf{M} = \begin{pmatrix} \mathbf{0} & -\mathbf{U} \\ -\mathbf{V}^\top & \mathbf{0} \end{pmatrix}, \ \mathbf{q} = \mathbf{1} \in \mathbb{R}^{m+n}.\end{gathered} \tag{B.21}$$

Provided that $\mathbf{I} - \mathbf{M}$ is nonsingular, this LCP is transformed into the AVE in Eq. (B.5), with $\mathbf{A} = (\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}+\mathbf{M})$ and $\mathbf{b} = (\mathbf{M}-\mathbf{I})^{-1}\mathbf{1}$. Invertibility can be ensured by a sufficiently small positive rescaling of both payoff matrices, which preserves Nash equilibria. The AVE solution corresponding to $\mathbf{w}_2 = \mathbf{x} + |\mathbf{x}| = 0$ must be excluded when recovering a Nash equilibrium. The resulting AVE need not satisfy the sufficient condition $\|\mathbf{A}^{-1}\|_2 < 1$. For example, when $\mathbf{U} = [1]$ and $\mathbf{V} = [\frac{1}{4}]$, $\mathbf{A}$ has eigenvalues 3 and $\frac{1}{3}$, implying $\|\mathbf{A}^{-1}\|_2 > 1$. Thus convergence guarantees relying on this condition do not apply to such cases.

C Fixed-point Iteration Methods for AVE

Concern the AVE $\mathbf{A}\mathbf{x} - |\mathbf{x}| = \mathbf{b}$ with a non-singular matrix $\mathbf{A}$. The most fundamental fixed point iteration has the form

$$\mathbf{x}^{(k+1)} = \mathbf{A}^{-1}(|\mathbf{x}^{(k)}| + \mathbf{b}), \ k = 0, 1, 2, \ldots \tag{C.1}$$

This method is also known as the Picard iteration (Rohn et al., 2014) and is guaranteed to converge if the spectral radius of the matrix $|\mathbf{A}^{-1}|$ is less than 1, *i.e.*, $\rho(|\mathbf{A}^{-1}|) < 1$.

Although the condition $\rho(|\mathbf{A}^{-1}|) < 1$ is also a sufficient condition for the AVE to have a unique solution, it is independent of the condition $\|\mathbf{A}^{-1}\|_2 < 1$. To address AVEs satisfying $\|\mathbf{A}^{-1}\|_2 < 1$, (Ke, 2020) proposed a new fixed-point iteration, defined as

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{A}^{-1}(\mathbf{y}^{(k)} + \mathbf{b}), \\ \mathbf{y}^{(k+1)} = (1 - \tau)\mathbf{y}^{(k)} + \tau|\mathbf{x}^{(k+1)}|, \end{cases} \quad k = 0, 1, 2, \ldots \tag{C.2}$$

The authors proved that with an appropriate parameter τ, this method converges for AVE instances with $\|\mathbf{A}^{-1}\|_2 < \frac{\sqrt{2}}{2}$.

The MFPI formulation in Eq. (6), proposed by Yu et al. (2022), introduces a nonsingular transformation $\mathbf{Q}$ and its inverse into the auxiliary-variable representation. Their analysis gives sufficient conditions covering $\|\mathbf{A}^{-1}\|_2 < 1$ through suitable scalar transformations, extending the range certified by the earlier analysis. This concerns sufficient bounds in the chosen state coordinates, rather than a change in the primal trajectory for a fixed relaxation parameter.

For fixed nonsingular $\mathbf{Q}$, setting $\mathbf{v}^{(k)} = \mathbf{Q}\mathbf{y}^{(k)}$ in Eq. (6) gives

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{A}^{-1}(\mathbf{v}^{(k)} + \mathbf{b}), \\ \mathbf{v}^{(k+1)} &= (1 - \tau)\mathbf{v}^{(k)} + \tau|\mathbf{x}^{(k+1)}|.\end{aligned} \tag{C.3}$$

Consequently, all choices of $\mathbf{Q}$ with the same τ and initial transformed state generate the same primal iterates in exact arithmetic. Eliminating $\mathbf{v}^{(k)}$ further yields

$$\mathbf{x}^{(k+2)} = (1 - \tau)\mathbf{x}^{(k+1)} + \tau\mathbf{A}^{-1}(|\mathbf{x}^{(k+1)}| + \mathbf{b}). \tag{C.4}$$

Under zero initialization, $\mathbf{x}^{(1)} = \mathbf{A}^{-1}\mathbf{b}$. The map in Eq. (C.4) has Lipschitz constant at most $|1-\tau| + \tau\nu$, where $\nu = \|\mathbf{A}^{-1}\|_2$. Thus $\nu < 1$ and $0 < \tau < 2/(1+\nu)$ provide a sufficient contraction condition independent of $\mathbf{Q}$, including $\mathbf{Q} = \mathbf{I}$. This is not a necessary condition for convergence on individual instances. Changing $\mathbf{Q}$ can affect coordinate-dependent error bounds and finite-precision implementations, but does not by itself imply faster convergence or greater numerical stability. When the linear operators are replaced by nonlinear learned modules, a change of auxiliary variables generally no longer eliminates these modules from the updates, even if they are exact inverses. Thus, the learned updates can differ from the classical recurrence.

Under certain conditions, the sequences $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ are guaranteed to converge to the solution $\mathbf{x}^*$ and $\mathbf{Q}^{-1}|\mathbf{x}^*|$, respectively. The following theorem formalizes this result.

Theorem C.1 *(Yu et al., 2022) Let $\mathbf{A}, \mathbf{Q} \in \mathbb{R}^{n\times n}$ be non-singular matrices and $\mathbf{b} \in \mathbb{R}^n$. Suppose the AVE $\mathbf{A}\mathbf{x} - |\mathbf{x}| = \mathbf{b}$ has at least one solution $\mathbf{x}^*$. Let $\mathbf{e}_k^{\mathbf{x}} := \mathbf{x}^{(k)} - \mathbf{x}^*$ and $\mathbf{e}_k^{\mathbf{y}} := \mathbf{y}^{(k)} - \mathbf{y}^*$ denote the iteration error. Denote*

$$\nu = \|\mathbf{A}^{-1}\|_2, \quad \alpha = \|\mathbf{Q}\|_2, \quad \beta = \|\mathbf{Q}^{-1}\|_2, \quad \mathbf{E}^{(k+1)} = \begin{pmatrix} \mathbf{e}_{k+1}^{\mathbf{x}} \\ \mathbf{e}_{k+1}^{\mathbf{y}} \end{pmatrix}. \tag{C.5}$$

We have

$$\|\mathbf{E}^{(k+1)}\|_2 \le \|\mathbf{R}\|_2 \|\mathbf{E}^{(k)}\|_2, \tag{C.6}$$

with $\|\mathbf{R}\|_2 = \sqrt{\alpha^2\nu^2 + (\alpha\beta\tau\nu + |1-\tau|)^2}$.

If

$$0 < \nu < \frac{1}{\alpha\sqrt{1+\beta^2}} \quad \text{and} \quad \frac{1-\sqrt{1-\alpha^2\nu^2}}{1-\alpha\beta\nu} < \tau < \frac{1+\sqrt{1-\alpha^2\nu^2}}{1+\alpha\beta\nu} \tag{C.7}$$

then we have

$$\|\mathbf{E}^{(k+1)}\|_2 < \|\mathbf{E}^{(k)}\|_2 \tag{C.8}$$

for $k = 0, 1, 2, \ldots$

Since $\alpha\beta \ge 1$, if $\nu \ge 1$ then $\alpha\beta\nu \ge 1$. For every $\tau > 0$,

$$\alpha\beta\tau\nu + |1-\tau| \ge \tau + |1-\tau| \ge 1. \tag{C.9}$$

Hence the displayed bound cannot satisfy $\|\mathbf{R}\|_2 < 1$. This only means that this sufficient condition cannot certify convergence when $\nu \ge 1$; it does not exclude convergent parameter choices for particular instances. The following lemma states an explicit Lipschitz bound in the chosen state coordinates for use in our approximation analysis.

D Theoretical analysis

D.1 Theorem 1

Lemma D.1 *Let $\mathbf{A}$ and $\mathbf{Q}$ be nonsingular and let $\tau > 0$. For any $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{2n}$, we have*

$$\|\Phi(\mathbf{z}_1) - \Phi(\mathbf{z}_2)\|_2 \le L(\tau, \alpha, \beta, \nu)\|\mathbf{z}_1 - \mathbf{z}_2\|_2, \tag{D.1}$$

where $L(\tau, \alpha, \beta, \nu) = \alpha\nu + |1-\tau| + \tau\alpha\beta\nu$, with $\alpha = \|\mathbf{Q}\|_2$, $\beta = \|\mathbf{Q}^{-1}\|_2$, and $\nu = \|\mathbf{A}^{-1}\|_2$. Especially, when

$$\nu < \frac{1}{\alpha(1+\beta)}, \frac{\alpha\nu}{1-\alpha\beta\nu} < \tau < \frac{2-\alpha\nu}{1+\alpha\beta\nu}, \tag{D.2}$$

we have $0 < L(\tau, \alpha, \beta, \nu) < 1$ and Φ is a continuous contraction mapping.

Proof. Write $\mathbf{z}_i = (\mathbf{x}_i^\top, \mathbf{y}_i^\top)^\top$ and $\Phi(\mathbf{z}_i) = ((\mathbf{x}_i^+)^\top, (\mathbf{y}_i^+)^\top)^\top$ for $i = 1, 2$. Let $\Delta\mathbf{y} = \mathbf{y}_1 - \mathbf{y}_2$, and define the other differences analogously. The first update gives

$$\Delta\mathbf{x}^+ = \mathbf{A}^{-1}\mathbf{Q}\Delta\mathbf{y}, \qquad \|\Delta\mathbf{x}^+\|_2 \le \alpha\nu\|\Delta\mathbf{y}\|_2.$$

For the second update, the triangle inequality and the nonexpansiveness of the entrywise absolute-value map imply, for $\tau > 0$,

$$\begin{aligned}\|\Delta\mathbf{y}^+\|_2 &\le |1-\tau|\|\Delta\mathbf{y}\|_2 + \tau\|\mathbf{Q}^{-1}\|_2\big\||\mathbf{x}_1^+| - |\mathbf{x}_2^+|\big\|_2 \\ &\le |1-\tau|\|\Delta\mathbf{y}\|_2 + \tau\beta\|\Delta\mathbf{x}^+\|_2 \\ &\le (|1-\tau| + \tau\alpha\beta\nu)\|\Delta\mathbf{y}\|_2.\end{aligned}$$

Combining the two block estimates yields

$$\begin{aligned}\|\Phi(\mathbf{z}_1) - \Phi(\mathbf{z}_2)\|_2 &\le \|\Delta\mathbf{x}^+\|_2 + \|\Delta\mathbf{y}^+\|_2 \\ &\le (\alpha\nu + |1-\tau| + \tau\alpha\beta\nu)\|\Delta\mathbf{y}\|_2 \\ &\le L(\tau,\alpha,\beta,\nu)\|\mathbf{z}_1 - \mathbf{z}_2\|_2.\end{aligned}$$

Thus $L(\tau,\alpha,\beta,\nu) < 1$ is a sufficient condition for Φ to be a contraction on $\mathbb{R}^{2n}$. The Banach fixed-point theorem (Ciesielski, 2007) then guarantees convergence to its unique fixed point.

We consider the following two cases based on the value of τ:

- Case 1: $\tau \le 1$. In this case, the condition $L(\tau,\alpha,\beta,\nu) < 1$ becomes $(1-\alpha\beta\nu)\tau > \alpha\nu$. Given $\tau > 0$, it is necessary that $\alpha\beta\nu < 1$ to avoid contradiction. Consequently, we obtain
$$\frac{\alpha\nu}{1-\alpha\beta\nu} < \tau \le 1.$$
To ensure the existence of a valid τ, we require $\frac{\alpha\nu}{1-\alpha\beta\nu} < 1$, which yields
$$\nu < \frac{1}{\alpha(1+\beta)}.$$
- Case 2: $\tau > 1$. In this case, the condition $L(\tau,\alpha,\beta,\nu) < 1$ simplifies to: $\tau < \frac{2-\alpha\nu}{1+\alpha\beta\nu}$. Similarly, $\frac{2-\alpha\nu}{1+\alpha\beta\nu} > 1$, from which we derive:
$$\tau < \frac{2-\alpha\nu}{1+\alpha\beta\nu}.$$

In summary, if $\nu < \frac{1}{\alpha(1+\beta)}$ and $\frac{\alpha\nu}{1-\alpha\beta\nu} < \tau < \frac{2-\alpha\nu}{1+\alpha\beta\nu}$, then Φ is a contraction mapping with $L(\tau,\alpha,\beta,\nu) < 1$. Under these conditions, the iteration converges to a unique fixed point, which concludes the proof. □

Proof of Theorem 1. Let $\mathbf{z}^*_{\mathrm{MFPI}}$ denote the optimal solution obtained via the MFPI iteration. For any $K \ge 1$:

$$\|\mathbf{z}^* - \mathbf{z}^*_\theta\|_2 \le \|\mathbf{z}^* - \mathbf{z}^{(K)}_{\mathrm{MFPI}}\|_2 + \|\mathbf{z}^{(K)}_{\mathrm{MFPI}} - \mathbf{z}^*_{\mathrm{MFPI}}\|_2 + \|\mathbf{z}^*_{\mathrm{MFPI}} - \mathbf{z}^*_\theta\|_2. \tag{D.3}$$

We now bound each of the three terms on the right-hand side. According to the results established in (Yu et al., 2022), we have:

$$\|\mathbf{z}^{(K)}_{\mathrm{MFPI}} - \mathbf{z}^*\|_2 \le \|\mathbf{R}\|_2\|\mathbf{z}^{(K-1)}_{\mathrm{MFPI}} - \mathbf{z}^*\|_2 \le \|\mathbf{R}\|_2^K\|\mathbf{z}^{(0)} - \mathbf{z}^*\|_2. \tag{D.4}$$

Recalling that Φ is a contraction mapping with Lipschitz constant $L < 1$ by Lemma D.1, the distance between the K-th iteration and the fixed point $\mathbf{z}^*_{\mathrm{MFPI}}$ can be bounded by the geometric series:

$$\begin{aligned}\|\mathbf{z}^*_{\mathrm{MFPI}} - \mathbf{z}^{(K)}_{\mathrm{MFPI}}\|_2 \le \sum_{i=K}^{\infty}\|\mathbf{z}^{(i+1)}_{\mathrm{MFPI}} - \mathbf{z}^{(i)}_{\mathrm{MFPI}}\|_2 &\le \sum_{i=K}^{\infty} L^i\|\mathbf{z}^{(1)}_{\mathrm{MFPI}} - \mathbf{z}^{(0)}_{\mathrm{MFPI}}\|_2 \\ &= \frac{L^K}{1-L}\|\Phi(\mathbf{z}^{(0)}) - \mathbf{z}^{(0)}\|_2.\end{aligned} \tag{D.5}$$

By the Universal Approximation Theorem (UAT) (Cybenko, 1989; Hornik et al., 1989), there exists a neural network f_θ that approximates Φ within a tolerance $\epsilon > 0$ such that $\|f_\theta(\mathbf{z}) - \Phi(\mathbf{z})\|_2 < \epsilon$. Consequently,

$$\begin{aligned}\|\mathbf{z}^*_\theta - \mathbf{z}^*_{\mathrm{MFPI}}\|_2 &\le \|\mathbf{z}^*_\theta - f_\theta(\mathbf{z}^*_\theta)\|_2 + \|f_\theta(\mathbf{z}^*_\theta) - \Phi(\mathbf{z}^*_\theta)\|_2 + \|\Phi(\mathbf{z}^*_\theta) - \Phi(\mathbf{z}^*_{\mathrm{MFPI}})\|_2 \\ &< \|\mathbf{z}^*_\theta - f_\theta(\mathbf{z}^*_\theta)\|_2 + \epsilon + L\|\mathbf{z}^*_\theta - \mathbf{z}^*_{\mathrm{MFPI}}\|_2,\end{aligned} \tag{D.6}$$

which yeilds:

$$\|\mathbf{z}^*_\theta - \mathbf{z}^*_{\mathrm{MFPI}}\|_2 < \frac{\epsilon}{1-L} + \frac{\|f_\theta(\mathbf{z}^*_\theta) - \mathbf{z}^*_\theta\|_2}{1-L}. \tag{D.7}$$

Combining Eq. (D.4), Eq. (D.5) and Eq. (D.7) concludes the proof. □

Corollary of Theorem 1 By further specifying the iteration with $f_{\mathbf{Q}}$, $f_{\mathbf{Q}^{-1}}$, and the sigmoid-based absolute-value module, we obtain the following refinement of Theorem 1. It decomposes the third term of Eq. (15) into the neural approximation tolerance and the individual module-level approximation errors.

Corollary D.1 *Under the assumptions of Theorem 1, fix $\epsilon > 0$ and a compact approximation domain $\mathcal{B} \subset \mathbb{R}^{2n}$. Let $\varepsilon_{\mathbf{Q}}$, $\varepsilon_{\mathbf{Q}^{-1}}$, and $\varepsilon_{\mathrm{abs}}$ denote uniform error bounds for the continuous learned modules $f_{\mathbf{Q}}$, $f_{\mathbf{Q}^{-1}}$, and the sigmoid-based absolute-value module, respectively, on the inputs induced by $\mathcal{B}$. Then, there exists a neural network f_θ such that, for any candidate solution $\mathbf{z}^*_\theta \in \mathcal{B}$, the error bound in* Eq. (15) *holds with its third term $\epsilon/(1-L)$ replaced by*

$$\frac{\epsilon + (1+\tau\|\mathbf{Q}^{-1}\|_2)\|\mathbf{A}^{-1}\|_2\varepsilon_{\mathbf{Q}} + \tau\varepsilon_{\mathbf{Q}^{-1}} + \tau\|\mathbf{Q}^{-1}\|_2\varepsilon_{\mathrm{abs}}}{1-L}. \tag{D.8}$$

Proof of Corollary D.1. The exact MFPI update is

$$\mathbf{x}^{\mathrm{new}} = \mathbf{A}^{-1}(\mathbf{Q}\mathbf{y} + \mathbf{b}), \qquad \mathbf{y}^{\mathrm{new}} = (1-\tau)\mathbf{y} + \tau\mathbf{Q}^{-1}|\mathbf{x}^{\mathrm{new}}|.$$

For a fixed update, suppress the layer index k in Eq. (7) and write $g(\mathbf{v}) = \varphi(\mathbf{v}) \odot \mathbf{v}$. Define the auxiliary learned-MFPI mapping by

$$\begin{aligned} \widetilde{\Phi}(\mathbf{z}) &= \begin{pmatrix} \tilde{\mathbf{x}}^{\mathrm{new}} \\ \tilde{\mathbf{y}}^{\mathrm{new}} \end{pmatrix}, \qquad \mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \\ \tilde{\mathbf{x}}^{\mathrm{new}} &= \mathbf{A}^{-1}\big(f_{\mathbf{Q}}(\mathbf{y}) + \mathbf{b}\big), \\ \tilde{\mathbf{y}}^{\mathrm{new}} &= (1-\tau)\mathbf{y} + \tau f_{\mathbf{Q}^{-1}}\big(g(\tilde{\mathbf{x}}^{\mathrm{new}})\big). \end{aligned} \tag{D.9}$$

All module errors are evaluated on inputs generated by $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathcal{B}$. More precisely, choose uniform bounds satisfying

$$\begin{aligned} \sup_{\mathbf{z}\in\mathcal{B}} \|f_{\mathbf{Q}}(\mathbf{y}) - \mathbf{Q}\mathbf{y}\|_2 &\le \varepsilon_{\mathbf{Q}}, \\ \sup_{\mathbf{z}\in\mathcal{B}} \|f_{\mathbf{Q}^{-1}}(g(\tilde{\mathbf{x}}^{\mathrm{new}})) - \mathbf{Q}^{-1}g(\tilde{\mathbf{x}}^{\mathrm{new}})\|_2 &\le \varepsilon_{\mathbf{Q}^{-1}}, \\ \sup_{\mathbf{z}\in\mathcal{B}} \|g(\tilde{\mathbf{x}}^{\mathrm{new}}) - |\tilde{\mathbf{x}}^{\mathrm{new}}|\|_2 &\le \varepsilon_{\mathrm{abs}}. \end{aligned} \tag{D.10}$$

These bounds are finite because the induced input sets are continuous images of the compact set $\mathcal{B}$. For brevity, set

$$\delta_{\mathrm{learn}} = (1+\tau\|\mathbf{Q}^{-1}\|_2)\|\mathbf{A}^{-1}\|_2\varepsilon_{\mathbf{Q}} + \tau\varepsilon_{\mathbf{Q}^{-1}} + \tau\|\mathbf{Q}^{-1}\|_2\varepsilon_{\mathrm{abs}}.$$

The mapping Eq. (D.9) replaces both linear operators and the absolute-value operation by their learned counterparts. It retains the sigmoid smoothing block and the internal nonlinearities of the learned modules, but does not include MFPI-Net's additional update activations, learnable residual step-size matrices, or channel expansion.

For $\mathbf{z} \in \mathcal{B}$, the error in the $\mathbf{x}$-component satisfies

$$\|\tilde{\mathbf{x}}^{\mathrm{new}} - \mathbf{x}^{\mathrm{new}}\|_2 \le \|\mathbf{A}^{-1}\|_2\varepsilon_{\mathbf{Q}}.$$

For the $\mathbf{y}$-component, write

$$\begin{aligned} \tilde{\mathbf{y}}^{\mathrm{new}} - \mathbf{y}^{\mathrm{new}} = \tau\big[f_{\mathbf{Q}^{-1}}(g(\tilde{\mathbf{x}}^{\mathrm{new}})) - \mathbf{Q}^{-1}g(\tilde{\mathbf{x}}^{\mathrm{new}})\big] \\ + \tau\mathbf{Q}^{-1}\big[g(\tilde{\mathbf{x}}^{\mathrm{new}}) - |\tilde{\mathbf{x}}^{\mathrm{new}}|\big] \\ + \tau\mathbf{Q}^{-1}\big[|\tilde{\mathbf{x}}^{\mathrm{new}}| - |\mathbf{x}^{\mathrm{new}}|\big]. \end{aligned}$$

Using Eq. (D.10) and the nonexpansiveness of the componentwise absolute-value map gives

$$\|\tilde{\mathbf{y}}^{\mathrm{new}} - \mathbf{y}^{\mathrm{new}}\|_2 \le \tau\varepsilon_{\mathbf{Q}^{-1}} + \tau\|\mathbf{Q}^{-1}\|_2\varepsilon_{\mathrm{abs}} + \tau\|\mathbf{Q}^{-1}\|_2\|\mathbf{A}^{-1}\|_2\varepsilon_{\mathbf{Q}}. \tag{D.11}$$

Combining both components by the triangle inequality yields

$$\sup_{\mathbf{z}\in\mathcal{B}} \|\widetilde{\Phi}(\mathbf{z}) - \Phi(\mathbf{z})\|_2 \le \delta_{\mathrm{learn}}. \tag{D.12}$$

The smoothing error can be made arbitrarily small within the specified sigmoid module. Indeed, for the logistic sigmoid $\sigma(t) = (1 + e^{-t})^{-1}$, choose one active subchannel in Eq. (7) with $\theta_{1,1} = 1$, $\theta_{1,2} = a > 0$, and $\theta_{1,3} = 0$, and set all other amplitudes to zero. Then $\varphi_a(t) = 2\sigma(at) - 1$, and

$$\big|t\varphi_a(t) - |t|\big| = \frac{2|t|}{1 + e^{a|t|}} \le 2|t|e^{-a|t|} \le \frac{2}{ae}.$$

Thus one may take $\varepsilon_{\mathrm{abs}} = 2\sqrt{n}/(ae) \to 0$ as $a \to \infty$.

Since the learned modules and g are continuous, $\widetilde{\Phi}$ is continuous on the compact set $\mathcal{B}$. By the universal approximation theorem (Cybenko, 1989; Hornik et al., 1989), a sufficiently expressive neural network with a linear output layer can satisfy

$$\sup_{\mathbf{z}\in\mathcal{B}} \|f_\theta(\mathbf{z}) - \widetilde{\Phi}(\mathbf{z})\|_2 < \epsilon. \tag{D.13}$$

Consequently,

$$\begin{aligned}\|f_\theta(\mathbf{z}^*_\theta) - \Phi(\mathbf{z}^*_\theta)\|_2 &\le \|f_\theta(\mathbf{z}^*_\theta) - \widetilde{\Phi}(\mathbf{z}^*_\theta)\|_2 + \|\widetilde{\Phi}(\mathbf{z}^*_\theta) - \Phi(\mathbf{z}^*_\theta)\|_2 \\ &< \epsilon + \delta_{\mathrm{learn}}.\end{aligned} \tag{D.14}$$

Substituting Eq. (D.14) into the contraction estimate Eq. (D.6) replaces ϵ by $\epsilon + \delta_{\mathrm{learn}}$ and gives the stated refinement of Eq. (15). The candidate residual is retained, so no fixed point or convergence of the approximating network is assumed. □

This refined bound describes how individual module errors propagate when approximating the classical operators and the absolute-value map. Approximation to a fixed pair $\mathbf{Q}, \mathbf{Q}^{-1}$ is distinct from inverse consistency: a small composition error alone does not imply small errors relative to either fixed operator. The principal theoretical motivation for the reconstruction loss in Eq. (11) is instead Theorem D.4, which bounds the AVE residual at an equilibrium of the associated dynamical system using absolute-value and composition errors. The empirical reconstruction loss encourages consistency on sampled states; it does not establish the uniform bounds assumed in that theorem.

D.2 Theorem 2

The following result gives a constructive approximation guarantee for the specific architecture in Algorithm 1, including its PReLU activations and final channel readout.

Theorem D.1 (Arbitrary-accuracy realization of finite MFPI trajectories) *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be nonsingular, let $\mathbf{Q} = q\mathbf{I}_n$ with $q \neq 0$, and fix $\tau > 0$. For any number of MFPI steps $N \ge 1$, channel width $d \ge 1$, and tolerance $\epsilon > 0$, there exist finite parameters of MFPI-Net such that its states approximate all the first N iterates of* Eq. (6) *to accuracy ϵ, starting from the common zero initialization. More precisely, write $\mathbf{x}_{\mathrm{M}}^{(k)}, \mathbf{y}_{\mathrm{M}}^{(k)}$ for the MFPI iterates and define*

$$E_k = \max\left\{ \frac{\|\mathbf{X}^{(k)} - \mathbf{x}_{\mathrm{M}}^{(k)}\mathbf{1}_{1\times d}\|_F}{\sqrt{d}}, \frac{|q|\|\mathbf{Y}^{(k)} - \mathbf{y}_{\mathrm{M}}^{(k)}\mathbf{1}_{1\times d}\|_F}{\sqrt{d}} \right\}. \tag{D.15}$$

Set $\nu = \|\mathbf{A}^{-1}\|_2$, $\rho = |1 - \tau| + \tau\nu$, and $S_N = \sum_{j=0}^{N-1} \rho^j$. A sufficient parameter choice is

$$\begin{gathered} K = N + 1, \qquad \mathbf{U}_{k,1} = \mathbf{I}_d, \qquad \mathbf{U}_{k,2} = \tau\mathbf{I}_d, \qquad \mathbf{w} = d^{-1}\mathbf{1}_{d\times 1}, \\ f_{\mathbf{Q}}(\mathbf{V}) = q\mathbf{V}, \qquad f_{\mathbf{Q}^{-1}}(\mathbf{V}) = q^{-1}\mathbf{V}, \\ C_\varphi = 1, \qquad (\theta^k_{1,1}, \theta^k_{1,2}, \theta^k_{1,3}) = (1, a, 0), \qquad a \ge \frac{2\tau\sqrt{n}\max\{1,\nu\}S_N}{e\epsilon}, \end{gathered} \tag{D.16}$$

with every PReLU negative-slope parameter set to 1. The two MLPs can realize the displayed maps using width d, zero biases, one weight matrix $q\mathbf{I}_d$ or $q^{-1}\mathbf{I}_d$, respectively, and identity matrices in any remaining affine layers. For these choices,

$$\max_{0\le k\le N} E_k \le \frac{2\tau\sqrt{n}\max\{1,\nu\}}{ea} S_N \le \epsilon, \qquad \|\mathbf{x}^{\mathrm{NN}} - \mathbf{x}_{\mathrm{M}}^{(N)}\|_2 \le \epsilon. \tag{D.17}$$

These estimates hold uniformly over $\mathbf{b} \in \mathbb{R}^n$.

Proof. With PReLU negative slopes equal to 1, all PReLU activations are identity maps. The stated MLP weights therefore realize the two scalar linear maps exactly and satisfy Eq. (10) exactly. The specified smoothing parameters give

$$\varphi_k(t) = 2\sigma(at) - 1, \qquad s_a(t) = t(2\sigma(at) - 1).$$

For every $t \in \mathbb{R}$,

$$0 \le |t| - s_a(t) = \frac{2|t|}{1 + e^{a|t|}} \le 2|t|e^{-a|t|} \le \frac{2}{ea}. \tag{D.18}$$

The last inequality follows by maximizing ue^{-au} over $u \ge 0$. Thus the entrywise vector map satisfies the global bound $\|s_a(\mathbf{v}) - |\mathbf{v}|\|_2 \le \delta_a$ with $\delta_a = 2\sqrt{n}/(ea)$. In particular, uniform approximation of the discontinuous sign function is unnecessary: it is the product $\varphi_k(t)t$ that approximates $|t|$ uniformly.

All channels are initially identical and remain identical under these parameters. Write $\mathbf{X}^{(k)} = \widetilde{\mathbf{x}}^{(k)}\mathbf{1}_{1\times d}$ and $\mathbf{Y}^{(k)} = \widetilde{\mathbf{y}}^{(k)}\mathbf{1}_{1\times d}$. Substitution into Eq. (12)–Eq. (13) yields exactly

$$\begin{aligned} \widetilde{\mathbf{x}}^{(k+1)} &= \mathbf{A}^{-1}(q\widetilde{\mathbf{y}}^{(k)} + \mathbf{b}), \\ \widetilde{\mathbf{y}}^{(k+1)} &= (1-\tau)\widetilde{\mathbf{y}}^{(k)} + \tau q^{-1} s_a(\widetilde{\mathbf{x}}^{(k+1)}). \end{aligned} \tag{D.19}$$

Here $\widehat{\mathbf{X}}^{(k)} = \mathbf{X}^{(k+1)}$ because the outer PReLU is the identity. Hence the only discrepancy from MFPI is the absolute-value surrogate.

Define $e_k^x = \|\widetilde{\mathbf{x}}^{(k)} - \mathbf{x}_{\mathrm{M}}^{(k)}\|_2$ and $e_k^y = |q|\|\widetilde{\mathbf{y}}^{(k)} - \mathbf{y}_{\mathrm{M}}^{(k)}\|_2$. Using Eq. (6), Eq. (D.19), the 1-Lipschitz property of the absolute-value map, and Eq. (D.18), we obtain

$$e_{k+1}^x \le \nu e_k^y, \qquad e_{k+1}^y \le |1-\tau|e_k^y + \tau(e_{k+1}^x + \delta_a) \le \rho e_k^y + \tau\delta_a.$$

Since $e_0^x = e_0^y = 0$, induction gives, with $S_0 = 0$,

$$e_k^y \le \tau\delta_a S_k, \qquad e_k^x \le \nu\tau\delta_a S_{k-1} \quad (k \ge 1).$$

Identical columns imply $E_k = \max\{e_k^x, e_k^y\}$. Because $S_k \le S_N$ for $k \le N$, this proves the state bound. Algorithm 1 performs $K-1$ updates, so $K = N+1$ produces $\mathbf{X}^{(N)}$ before readout. The selected averaging vector gives $\mathbf{x}^{\mathrm{NN}} = \mathbf{X}^{(N)}\mathbf{w} = \widetilde{\mathbf{x}}^{(N)}$, proving the output bound. None of these error estimates depends on $\mathbf{b}$. □

D.3 THEOREM 3.

Proof of Theorem 3. Fix the problem data and all layer parameters. Let L_1, L_2 be Frobenius-norm Lipschitz constants of $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$, respectively. By the boundedness assumption, choose $R_{\mathbf{X}} < \infty$ such that $|\widehat{\mathbf{X}}^{(k)}(\mathbf{Z})| \le R_{\mathbf{X}}$ entrywise for all $\mathbf{Z} \in \mathcal{S}_k$. Define

$$\begin{gathered} H_k = B_k + R_{\mathbf{X}}G_k, \qquad B_k = \sum_{l=1}^{C_\varphi} |\theta_{l,1}^k|, \qquad G_k = \frac{1}{2}\sum_{l=1}^{C_\varphi} |\theta_{l,1}^k\theta_{l,2}^k|, \\ a_k = \|\mathbf{I}_d - \mathbf{U}_{k,1}\|_2, \qquad b_k = \nu L_1\|\mathbf{U}_{k,1}\|_2, \\ c_k = \|\mathbf{I}_d - \mathbf{U}_{k,2}\|_2, \qquad d_k = L_2 H_k\|\mathbf{U}_{k,2}\|_2, \qquad \nu = \|\mathbf{A}^{-1}\|_2. \end{gathered} \tag{D.20}$$

We first bound the scalar smoothing map $g_k(x) = \varphi_k(x)x$. From Eq. (7) and $0 < \sigma(u) < 1$, $\sigma'(u) \le 1/4$, we obtain

$$|\varphi_k(x)| \le B_k, \qquad |\varphi_k'(x)| \le G_k.$$

Hence $|g_k'(x)| \le B_k + R_{\mathbf{X}}G_k = H_k$ whenever $|x| \le R_{\mathbf{X}}$. The interval $[-R_{\mathbf{X}}, R_{\mathbf{X}}]$ contains the segment between each pair of corresponding entries. Applying the scalar mean value theorem entrywise and summing squared differences yields

$$\|g_k(\widehat{\mathbf{X}}_1) - g_k(\widehat{\mathbf{X}}_2)\|_F \le H_k\|\widehat{\mathbf{X}}_1 - \widehat{\mathbf{X}}_2\|_F. \tag{D.21}$$

For $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathcal{S}_k$, write $\Delta\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_2$ and define the other differences analogously, suppressing the layer index on the states. Using $\|\mathbf{MVN}\|_F \le \|\mathbf{M}\|_2\|\mathbf{V}\|_F\|\mathbf{N}\|_2$, the residual $\mathbf{X}$ equation gives

$$\begin{aligned} \Delta\widehat{\mathbf{X}} &= \Delta\mathbf{X}(\mathbf{I}_d - \mathbf{U}_{k,1}) + \mathbf{A}^{-1}\big(f_{\mathbf{Q}}(\mathbf{Y}_1) - f_{\mathbf{Q}}(\mathbf{Y}_2)\big)\mathbf{U}_{k,1}, \\ \|\Delta\widehat{\mathbf{X}}\|_F &\le a_k\|\Delta\mathbf{X}\|_F + b_k\|\Delta\mathbf{Y}\|_F. \end{aligned} \tag{D.22}$$

Similarly, the residual $\mathbf{Y}$ equation and Eq. (D.21) imply

$$\begin{aligned}\|\Delta\widehat{\mathbf{Y}}\|_F &\le c_k\|\Delta\mathbf{Y}\|_F + \|\mathbf{U}_{k,2}\|_2 L_2 H_k \|\Delta\widehat{\mathbf{X}}\|_F \\ &\le d_k a_k \|\Delta\mathbf{X}\|_F + (c_k + d_k b_k)\|\Delta\mathbf{Y}\|_F. \end{aligned} \tag{D.23}$$

Therefore, componentwise in the two block norms,

$$\begin{pmatrix} \|\Delta\widehat{\mathbf{X}}\|_F \\ \|\Delta\widehat{\mathbf{Y}}\|_F \end{pmatrix} \le \begin{pmatrix} a_k & b_k \\ d_k a_k & c_k + d_k b_k \end{pmatrix} \begin{pmatrix} \|\Delta\mathbf{X}\|_F \\ \|\Delta\mathbf{Y}\|_F \end{pmatrix}.$$

To include the outer activations, let $\eta_{k,X,j}$ and $\eta_{k,Y,j}$ denote their negative-slope parameters, where j ranges over the entries or channels to which distinct parameters are assigned. Define

$$p_{k,X} = \max\{1, \max_j |\eta_{k,X,j}|\}, \qquad p_{k,Y} = \max\{1, \max_j |\eta_{k,Y,j}|\}.$$

The scalar map $\mathrm{PReLU}_\eta(t) = \max\{t, 0\} + \eta \min\{t, 0\}$ is Lipschitz with constant $\max\{1, |\eta|\}$. Applying this bound entrywise gives

$$\begin{pmatrix} \|\Delta\mathbf{X}^{(k+1)}\|_F \\ \|\Delta\mathbf{Y}^{(k+1)}\|_F \end{pmatrix} \le \begin{pmatrix} p_{k,X} a_k & p_{k,X} b_k \\ p_{k,Y} d_k a_k & p_{k,Y}(c_k + d_k b_k) \end{pmatrix} \begin{pmatrix} \|\Delta\mathbf{X}\|_F \\ \|\Delta\mathbf{Y}\|_F \end{pmatrix}.$$

All entries of this comparison matrix are nonnegative. Taking the Euclidean norm of the two-vectors and using $\|\Delta\mathbf{Z}\|_F^2 = \|\Delta\mathbf{X}\|_F^2 + \|\Delta\mathbf{Y}\|_F^2$ proves Eq. (16) with

$$\begin{aligned}\Gamma_k &= \left\| \begin{pmatrix} p_{k,X} a_k & p_{k,X} b_k \\ p_{k,Y} d_k a_k & p_{k,Y}(c_k + d_k b_k) \end{pmatrix} \right\|_2 \\ &\le \sqrt{p_{k,X}^2 (a_k^2 + b_k^2) + p_{k,Y}^2 \big((d_k a_k)^2 + (c_k + d_k b_k)^2\big)}. \end{aligned} \tag{D.24}$$

The last inequality bounds the spectral norm by the Frobenius norm. No restriction on the finite learned slopes is needed. If all outer PReLU slopes have absolute value at most 1, then $p_{k,X} = p_{k,Y} = 1$ and the activations are nonexpansive. The Lipschitz constants L_1, L_2 already account for the activations inside the two MLPs. □

D.4 Theorem 4

We establish convergence of the continuous MFPI dynamics and then combine it with equilibrium consistency and finite-depth trajectory approximation. Throughout this subsection, all matrix norms are Frobenius norms unless the spectral norm is explicitly indicated. Write $s(\mathbf{X}) = \varphi(\mathbf{X}) \odot \mathbf{X}$ and $\mathbf{Z} = (\mathbf{X}^\top, \mathbf{Y}^\top)^\top$. For the global results below, the state space in Eq. (17) is $\Omega = \mathbb{R}^{2n \times d}$, and all learned modules are fixed.

Theorem D.2 (Global exponential convergence of the MFPI dynamics) *Let* $\mathbf{A}$ *be nonsingular. Suppose that* $f_{\mathbf{Q}}$, $f_{\mathbf{Q}^{-1}}$, *and* s *are globally Lipschitz, with positive Lipschitz upper bounds* L_Q, $L_{Q^{-1}}$, *and* L_s, *respectively. Define*

$$r = \|\mathbf{A}^{-1}\|_2, \qquad a = rL_Q, \qquad b = L_{Q^{-1}} L_s, \qquad \rho = \sqrt{ab} < 1. \tag{D.25}$$

For a pair of matrices, introduce the weighted norm

$$\|(\mathbf{U}, \mathbf{V})\|_w^2 = \|\mathbf{U}\|_F^2 + \frac{a}{b}\|\mathbf{V}\|_F^2. \tag{D.26}$$

Then Eq. (17) *has a unique equilibrium* $\mathbf{Z}_{\mathrm{eq}} = (\mathbf{X}_{\mathrm{eq}}^\top, \mathbf{Y}_{\mathrm{eq}}^\top)^\top$, *and every initial state generates a unique global solution satisfying*

$$\|\mathbf{Z}(t) - \mathbf{Z}_{\mathrm{eq}}\|_w \le e^{-(1-\rho)t}\|\mathbf{Z}(0) - \mathbf{Z}_{\mathrm{eq}}\|_w, \qquad t \ge 0. \tag{D.27}$$

Thus the equilibrium is globally exponentially stable.

Proof. Define the equilibrium map

$$\mathcal{M}(\mathbf{X},\mathbf{Y}) = \big(\mathbf{A}^{-1}(f_{\mathbf{Q}}(\mathbf{Y})+\mathbf{B}), f_{\mathbf{Q}^{-1}}(s(\mathbf{X}))\big). \tag{D.28}$$

For $\Delta\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_2$ and $\Delta\mathbf{Y} = \mathbf{Y}_1 - \mathbf{Y}_2$, the Lipschitz assumptions imply

$$\begin{aligned}\|\mathcal{M}(\mathbf{Z}_1) - \mathcal{M}(\mathbf{Z}_2)\|_w^2 &\le a^2\|\Delta\mathbf{Y}\|_F^2 + \frac{a}{b}b^2\|\Delta\mathbf{X}\|_F^2 \\ &= \rho^2\|\mathbf{Z}_1 - \mathbf{Z}_2\|_w^2.\end{aligned}$$

The weighted state space is complete, so the Banach fixed-point theorem gives a unique fixed point of $\mathcal{M}$. Since the vector field is $G(\mathbf{Z}) = \mathcal{M}(\mathbf{Z}) - \mathbf{Z}$, this fixed point is precisely the unique equilibrium. Global Lipschitz continuity of G also gives existence and uniqueness of solutions for all $t \ge 0$.

Let $\langle\cdot,\cdot\rangle_w$ denote the inner product associated with Eq. (D.26), and set $V(\mathbf{Z}) = \frac{1}{2}\|\mathbf{Z} - \mathbf{Z}_{\mathrm{eq}}\|_w^2$. Along any solution,

$$\begin{aligned}\dot{V} &= \langle \mathbf{Z} - \mathbf{Z}_{\mathrm{eq}}, \mathcal{M}(\mathbf{Z}) - \mathcal{M}(\mathbf{Z}_{\mathrm{eq}}) - (\mathbf{Z} - \mathbf{Z}_{\mathrm{eq}})\rangle_w \\ &\le -(1-\rho)\|\mathbf{Z} - \mathbf{Z}_{\mathrm{eq}}\|_w^2 = -2(1-\rho)V.\end{aligned}$$

Integrating this inequality proves Eq. (D.27). □

Condition Eq. (D.25) is a sufficient condition on the learned modules, namely $rL_QL_{Q^{-1}}L_s < 1$; it does not follow from approximate mutual inversion alone. Positive upper bounds may be chosen even when a module is constant, so the weighted norm is well defined. In the special case $f_{\mathbf{Q}}(\mathbf{Y}) = \mathbf{QY}$ and $f_{\mathbf{Q}^{-1}}(\mathbf{U}) = \mathbf{Q}^{-1}\mathbf{U}$ with nonsingular $\mathbf{Q}$, the change of variables $\mathbf{W} = \mathbf{QY}$ gives

$$\dot{\mathbf{X}} = -\mathbf{X} + \mathbf{A}^{-1}(\mathbf{W}+\mathbf{B}), \qquad \dot{\mathbf{W}} = -\mathbf{W} + s(\mathbf{X}).$$

The same proof applies with $a = r$ and $b = L_s$, so $rL_s < 1$ suffices in this case. In particular, for $s(\mathbf{X}) = |\mathbf{X}|$, the condition $r < 1$ ensures exponential convergence without a condition-number restriction on $\mathbf{Q}$.

Theorem D.3 (Trajectory approximation by MFPI-Net) *With suitable parameters, MFPI-Net approximates the continuous MFPI dynamics in* Eq. (17)*. Specifically, let* $\mathbf{A}$ *be nonsingular, fix* $T > 0$*, and set* $h = T/K$*. Write* $\mathbf{z}^k = \mathrm{vec}(\mathbf{X}^{(k)}, \mathbf{Y}^{(k)})$*, where* K *counts update layers and excludes the final channel readout. With the same initial state* $\mathbf{z}_0$*, the trajectory* $\mathbf{z}(T)$ *and the state* $\mathbf{z}^K$ *after* K *updates of* Eq. (18) *satisfy*

$$\|\mathbf{z}(T) - \mathbf{z}^K\|_2 \le \frac{CT}{K}\,\frac{e^{LT}-1}{L}, \tag{D.29}$$

where $C \ge 0$ *and* $L > 0$ *are constants independent of* K *and* h *for fixed* T*, system and network parameters, and initialization. Thus the trajectory error is* $O(K^{-1})$*.*

Proof of Theorem D.3 Write $\mathbf{z} = \mathrm{vec}(\mathbf{X},\mathbf{Y})$ and let $\mathbf{z}^K$ denote the state before the final channel readout. We make the suitable parameter choices explicit. Set every negative-slope parameter of the PReLU activations in both update layers and in the MLP blocks $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$ to 1. Indeed, $\mathrm{PReLU}_\alpha(u) = \max\{u,0\} + \alpha\min\{u,0\}$ satisfies $\mathrm{PReLU}_1(u) = u$, so these activations are exactly identity maps. For every update layer k, choose $\mathbf{U}_{k,1} = \mathbf{U}_{k,2} = h\mathbf{I}_d$ with $h = T/K$ and $\varphi_k = \varphi$, and use the same fixed blocks $f_{\mathbf{Q}}, f_{\mathbf{Q}^{-1}}$ as in Eq. (17). The smoothing operator $s(\mathbf{X}) = \varphi(\mathbf{X}) \odot \mathbf{X}$ is retained. All block weights, biases, and smoothing parameters are fixed and finite. We use the following explicit constants to verify regularity, boundedness, and the discretization error:

$$\begin{aligned}
&a_{\mathbf{Q}} = \prod_l \|\mathbf{W}_{\mathbf{Q}}^l\|_2, \qquad a_{\mathbf{Q}^{-1}} = \prod_l \|\mathbf{W}_{\mathbf{Q}^{-1}}^l\|_2, \qquad \nu = \|\mathbf{A}^{-1}\|_2, \\
&P_\varphi = \sum_{l=1}^{C_\varphi} |\theta_{l,1}|, \qquad S_\varphi = P_\varphi + \sum_{l=1}^{C_\varphi} |\theta_{l,1}|\left(2 + \frac{|\theta_{l,3}|}{2}\right), \\
&L_1 = \sqrt{1+\nu^2 a_{\mathbf{Q}}^2}, \qquad L_2 = \sqrt{1 + a_{\mathbf{Q}^{-1}}^2 S_\varphi^2}, \qquad L = L_1 + L_2, \\
&c_1 = \|\mathbf{A}^{-1}(f_{\mathbf{Q}}(0)+\mathbf{B})\|_F, \qquad c_2 = \|f_{\mathbf{Q}^{-1}}(0)\|_F, \qquad c = c_1 + c_2, \\
&D = \max\{c_1/L_1, c_2/L_2\}, \qquad R = e^{LT}(\|\mathbf{z}_0\|_2 + D), \\
&C_1 = L_1R + c_1, \qquad M = L(LR+c), \\
&q_h = (1+hL_1)(1+hL_2), \qquad C = M/2 + L_2C_1.
\end{aligned} \tag{D.30}$$

Here the products run over all affine layers of the corresponding blocks.

Regularity of the specific vector field. With the PReLU parameters chosen as above, both MLP blocks are compositions of affine maps. Matrix multiplication and the Frobenius norm therefore give

$$\|f_{\mathbf{Q}}(\mathbf{U}) - f_{\mathbf{Q}}(\mathbf{V})\|_F \le a_{\mathbf{Q}}\|\mathbf{U}-\mathbf{V}\|_F, \qquad \|f_{\mathbf{Q}^{-1}}(\mathbf{U}) - f_{\mathbf{Q}^{-1}}(\mathbf{V})\|_F \le a_{\mathbf{Q}^{-1}}\|\mathbf{U}-\mathbf{V}\|_F.$$

To bound the smoothing operator globally, write $\sigma(u) = (1+e^{-u})^{-1}$. The elementary bounds $\sigma'(u) \le 1/4$ and $|u|\sigma'(u) \le |u|e^{-|u|} \le 1$ imply, for any real a, b, x,

$$|ax|\sigma'(ax \pm b) \le |ax \pm b|\sigma'(ax \pm b) + |b|\sigma'(ax \pm b) \le 1 + |b|/4.$$

Consequently, Eq. (7) yields $|\varphi(x)| \le P_\varphi$ and

$$|s'(x)| = |\varphi(x) + x\varphi'(x)| \le S_\varphi, \qquad \|s(\mathbf{U}) - s(\mathbf{V})\|_F \le S_\varphi\|\mathbf{U}-\mathbf{V}\|_F.$$

This bound also covers zero slope parameters, without dividing by them. For $\mathbf{z} = \mathrm{vec}(\mathbf{X}, \mathbf{Y})$, define $F_1(\mathbf{z}) = \mathrm{vec}(G_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}), 0)$ and $F_2(\mathbf{z}) = \mathrm{vec}(0, G_{\mathbf{Y}}(\mathbf{X}, \mathbf{Y}))$. For $\mathbf{u} = \mathrm{vec}(\mathbf{X}, \mathbf{Y})$ and $\mathbf{v} = \mathrm{vec}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}})$,

$$\|F_1(\mathbf{u}) - F_1(\mathbf{v})\|_2 \le \|\mathbf{X} - \widehat{\mathbf{X}}\|_F + \nu a_{\mathbf{Q}}\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F \le L_1\|\mathbf{u}-\mathbf{v}\|_2,$$
$$\|F_2(\mathbf{u}) - F_2(\mathbf{v})\|_2 \le \|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F + a_{\mathbf{Q}^{-1}}S_\varphi\|\mathbf{X} - \widehat{\mathbf{X}}\|_F \le L_2\|\mathbf{u}-\mathbf{v}\|_2.$$

Thus $F = F_1 + F_2$ is globally Lipschitz with constant L and is smooth. In particular, the ODE admits a unique global solution. Since $s(0) = 0$, $\|F_i(0)\|_2 = c_i$, and hence

$$\|F_i(\mathbf{z})\|_2 \le L_i\|\mathbf{z}\|_2 + c_i \le L_i(\|\mathbf{z}\|_2 + D), \qquad i = 1, 2. \tag{D.31}$$

A common bounded region for the trajectory and substeps. The integral equation for the ODE and Eq. (D.31), followed by Gronwall's inequality, give

$$\|\mathbf{z}(t)\|_2 + D \le e^{Lt}(\|\mathbf{z}_0\|_2 + D), \qquad 0 \le t \le T.$$

For either Euler substep $E_i(\mathbf{z}) = \mathbf{z} + hF_i(\mathbf{z})$, the same growth estimate gives

$$\|E_i(\mathbf{z})\|_2 + D \le (1 + hL_i)(\|\mathbf{z}\|_2 + D).$$

By Eq. (18), $E_1(\mathbf{z}^k) = \mathrm{vec}(\mathbf{X}^{(k+1)}, \mathbf{Y}^{(k)})$ and $\mathbf{z}^{k+1} = E_2(E_1(\mathbf{z}^k))$. Therefore,

$$\|\mathbf{z}^k\|_2 + D \le q_h^k(\|\mathbf{z}_0\|_2 + D), \qquad \|E_1(\mathbf{z}^k)\|_2 + D \le (1 + hL_1)q_h^k(\|\mathbf{z}_0\|_2 + D).$$

Because $q_h^K \le e^{LT}$ and $(1+hL_1)q_h^k \le q_h^K$ for $0 \le k < K$, all network states and intermediate states lie in the closed ball $\Omega_T = \{\mathbf{z} : \|\mathbf{z}\|_2 \le R\}$. The same is true of $E_1(\mathbf{z}(t_k))$, where $t_k = kh$ and $k < K$, since $(1 + hL_1)e^{Lt_k} \le e^{L(t_k+h)} \le e^{LT}$. Thus the common bounded region is established rather than assumed. On this region, Eq. (D.31) gives $\|F_1(\mathbf{z})\|_2 \le C_1$ and $\|F(\mathbf{z})\|_2 \le LR + c$. Moreover, global Lipschitz continuity and smoothness imply $\|DF(\mathbf{z})\|_2 \le L$, so along the exact trajectory

$$\left\| \frac{d}{dt} F(\mathbf{z}(t)) \right\|_2 = \|DF(\mathbf{z}(t))F(\mathbf{z}(t))\|_2 \le L(LR + c) = M. \tag{D.32}$$

This verifies the derivative bound as well as the bounded-field condition.

Exact network realization and one-step error. Substituting $\mathbf{U}_{k,1} = \mathbf{U}_{k,2} = h\mathbf{I}_d$ and $\varphi_k = \varphi$ into Eq. (12)–Eq. (13), with the PReLU parameters chosen as above, gives exactly Eq. (18), or equivalently $\mathbf{z}^{k+1} = E_2(E_1(\mathbf{z}^k))$. There is consequently no vector-field approximation error. The common one-layer map is

$$S_h(\mathbf{z}) = E_2(E_1(\mathbf{z})) = \mathbf{z} + hF_1(\mathbf{z}) + hF_2(\mathbf{z} + hF_1(\mathbf{z})). \tag{D.33}$$

The Lipschitz bounds already proved imply

$$\|S_h(\mathbf{u}) - S_h(\mathbf{v})\|_2 \le q_h\|\mathbf{u}-\mathbf{v}\|_2. \tag{D.34}$$

Integrating Eq. (D.32) over one time step yields

$$\mathbf{z}(t_{k+1}) = \mathbf{z}(t_k) + hF(\mathbf{z}(t_k)) + \mathbf{r}_k, \qquad \|\mathbf{r}_k\|_2 \le Mh^2/2. \tag{D.35}$$

Also, for $\mathbf{z} = \mathbf{z}(t_k)$,

$$\|S_h(\mathbf{z}) - \mathbf{z} - hF(\mathbf{z})\|_2 = h\|F_2(\mathbf{z} + hF_1(\mathbf{z})) - F_2(\mathbf{z})\|_2 \le h^2 L_2 C_1.$$

Combining these estimates proves the local error bound

$$\|\mathbf{z}(t_{k+1}) - S_h(\mathbf{z}(t_k))\|_2 \le Ch^2. \tag{D.36}$$

Accumulation over the network layers. Set $e_k = \|\mathbf{z}(t_k) - \mathbf{z}^k\|_2$. The stability and local error estimates give

$$e_{k+1} \le q_h e_k + Ch^2, \qquad e_0 = 0. \tag{D.37}$$

Since $q_h - 1 = h(L_1 + L_2 + hL_1L_2) > 0$, summing the geometric series yields

$$e_K \le Ch^2 \sum_{j=0}^{K-1} q_h^j = Ch \frac{q_h^K - 1}{L_1 + L_2 + hL_1L_2}.$$

Finally, $q_h^K \le e^{LT}$ and $L_1 + L_2 + hL_1L_2 \ge L$ prove Eq. (D.29). The constants C and L are independent of K and h. For a fixed channel readout $\mathbf{w}$, the output error is also $O(K^{-1})$, since $\|(\mathbf{X}(T) - \mathbf{X}^{(K)})\mathbf{w}\|_2 \le \|\mathbf{X}(T) - \mathbf{X}^{(K)}\|_F \|\mathbf{w}\|_2$.

□

Theorem D.4 (Equilibrium consistency) *Let $(\mathbf{X}_{\mathrm{eq}}, \mathbf{Y}_{\mathrm{eq}})$ be an equilibrium of* Eq. (17)*. Suppose that the learned absolute-value operator and the learned inverse pair satisfy*

$$\|\varphi(\mathbf{X}) \odot \mathbf{X} - |\mathbf{X}|\| \le \varepsilon_1, \quad \|f_{\mathbf{Q}}(f_{\mathbf{Q}^{-1}}(\mathbf{X})) - \mathbf{X}\| \le \varepsilon_2 \tag{D.38}$$

on compact sets containing $\mathbf{X}_{\mathrm{eq}}$ and $\varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}}$, respectively. Then

$$\|\mathbf{A}\mathbf{X}_{\mathrm{eq}} - |\mathbf{X}_{\mathrm{eq}}| - \mathbf{B}\| \le \varepsilon_1 + \varepsilon_2. \tag{D.39}$$

In particular, if $\varphi(\mathbf{X}) \odot \mathbf{X} = |\mathbf{X}|$ and $f_{\mathbf{Q}} \circ f_{\mathbf{Q}^{-1}}$ is the identity on the Ω, each channel of $\mathbf{X}_{\mathrm{eq}}$ is an exact solution of the AVE.

Proof of Theorem D.4 Since $(\mathbf{X}_{\mathrm{eq}}, \mathbf{Y}_{\mathrm{eq}})$ is an equilibrium point of Eq. (17),

$$\mathbf{A}\mathbf{X}_{\mathrm{eq}} - f_{\mathbf{Q}}(\mathbf{Y}_{\mathrm{eq}}) - \mathbf{B} = 0, \qquad \mathbf{Y}_{\mathrm{eq}} = f_{\mathbf{Q}^{-1}}(\varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}}). \tag{D.40}$$

Substituting $\mathbf{Y}_{\mathrm{eq}}$ into the first equality in Eq. (D.40) yields

$$\|\mathbf{A}\mathbf{X}_{\mathrm{eq}} - |\mathbf{X}_{\mathrm{eq}}| - \mathbf{B}\| \le \|f_{\mathbf{Q}}(f_{\mathbf{Q}^{-1}}(\varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}})) - \varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}}\| + \|\varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}} - |\mathbf{X}_{\mathrm{eq}}|\|$$

Combining Eq. (D.38) completes the proof. □

Theorem D.5 (Full version of Theorem 4) *Assume the hypotheses of Theorem D.2 and, in addition, $\|\mathbf{A}^{-1}\|_2 < 1$. Suppose that at its unique equilibrium,*

$$\|\varphi(\mathbf{X}_{\mathrm{eq}}) \odot \mathbf{X}_{\mathrm{eq}} - |\mathbf{X}_{\mathrm{eq}}|\|_F \le \varepsilon_1, \quad \|f_{\mathbf{Q}}(f_{\mathbf{Q}^{-1}}(s(\mathbf{X}_{\mathrm{eq}}))) - s(\mathbf{X}_{\mathrm{eq}})\|_F \le \varepsilon_2, \tag{D.41}$$

where $\varepsilon_1, \varepsilon_2 \ge 0$. Let $\mathbf{x}^*$ be the unique solution of $\mathbf{A}\mathbf{x} - |\mathbf{x}| = \mathbf{b}$ and set $\mathbf{X}^* = \mathbf{x}^* \mathbf{1}_{1\times d}$. Fix $T > 0$ and an initial state $\mathbf{Z}_0$. For each integer $K \ge 1$, set $h = T/K$ and use the sequential updates in Eq. (18) *from $\mathbf{Z}_0$. These are the MFPI-Net updates with the negative-slope parameters of the outer PReLU activations set to 1, $\mathbf{U}_{k,1} = \mathbf{U}_{k,2} = h\mathbf{I}_d$, and the same fixed modules at every layer, including $\varphi_k = \varphi$. Then there exist constants $L > 0$ and $C \ge 0$, independent of K for fixed T, modules, problem data, and initialization, such that*

$$\|\mathbf{X}^{(K)} - \mathbf{X}^*\|_F \le \underbrace{\frac{CT}{K}\frac{e^{LT} - 1}{L}}_{\text{trajectory discretization}} + \underbrace{e^{-(1-\rho)T} E_0}_{\text{distance to equilibrium}} + \underbrace{\frac{r}{1-r}(\varepsilon_1 + \varepsilon_2)}_{\text{module inconsistency}}, \tag{D.42}$$

where $E_0 = \|\mathbf{Z}_0 - \mathbf{Z}_{\mathrm{eq}}\|_w$. Here $\mathbf{X}^{(K)}$ is the state before the final channel readout.

Proof. First, the map $\mathbf{X} \mapsto \mathbf{A}^{-1}(|\mathbf{X}| + \mathbf{B})$ is a contraction in the Frobenius norm with constant $r < 1$, since entrywise absolute value is nonexpansive. It therefore has a unique fixed point $\mathbf{X}^*$. Its columns coincide with the unique solution $\mathbf{x}^*$ of the original AVE because every column has the same right-hand side $\mathbf{b}$.

At equilibrium, $\mathbf{Y}_{\mathrm{eq}} = f_{\mathbf{Q}^{-1}}(s(\mathbf{X}_{\mathrm{eq}}))$ and $\mathbf{A}\mathbf{X}_{\mathrm{eq}} = f_{\mathbf{Q}}(\mathbf{Y}_{\mathrm{eq}}) + \mathbf{B}$. Consequently, Theorem D.4, or directly the triangle inequality in Eq. (D.41), gives

$$\|\mathbf{R}_{\mathrm{eq}}\|_F \le \varepsilon_1 + \varepsilon_2, \qquad \mathbf{R}_{\mathrm{eq}} = \mathbf{A}\mathbf{X}_{\mathrm{eq}} - |\mathbf{X}_{\mathrm{eq}}| - \mathbf{B}.$$

Subtracting the equation for $\mathbf{X}^*$ yields

$$\begin{aligned}\|\mathbf{X}_{\mathrm{eq}} - \mathbf{X}^*\|_F &\le r\big(\||\mathbf{X}_{\mathrm{eq}}| - |\mathbf{X}^*|\|_F + \|\mathbf{R}_{\mathrm{eq}}\|_F\big) \\ &\le r\|\mathbf{X}_{\mathrm{eq}} - \mathbf{X}^*\|_F + r(\varepsilon_1 + \varepsilon_2).\end{aligned}$$

Hence

$$\|\mathbf{X}_{\mathrm{eq}} - \mathbf{X}^*\|_F \le \frac{r}{1-r}(\varepsilon_1 + \varepsilon_2). \tag{D.43}$$

Theorem D.2 also gives

$$\|\mathbf{X}(T) - \mathbf{X}_{\mathrm{eq}}\|_F \le \|\mathbf{Z}(T) - \mathbf{Z}_{\mathrm{eq}}\|_w \le e^{-(1-\rho)T}E_0. \tag{D.44}$$

For completeness, the trajectory bound of Theorem D.3 holds under the global Lipschitz assumptions above, without requiring differentiability of the internal network blocks. In the ordinary Euclidean norm of the vectorized state, write $F_1 = (G_{\mathbf{X}}, 0)$ and $F_2 = (0, G_{\mathbf{Y}})$. They have Lipschitz upper bounds $L_1 = \sqrt{1+a^2}$ and $L_2 = \sqrt{1+b^2}$, respectively; set $L = L_1 + L_2$. Along the exact trajectory on $[0, T]$, define the finite constants

$$M = \max_{0\le t\le T} \|G(\mathbf{z}(t))\|_2, \qquad M_1 = \max_{0\le t\le T} \|F_1(\mathbf{z}(t))\|_2, \qquad C = \tfrac{1}{2}LM + L_2M_1.$$

Let $S_h = (I + hF_2) \circ (I + hF_1)$ be one sequential step. Its Lipschitz constant is at most $(1 + hL_1)(1 + hL_2) \le e^{Lh}$. The integral equation for the exact solution gives, for $t, t + h \in [0, T]$,

$$\|\mathbf{z}(t+h) - \mathbf{z}(t) - hG(\mathbf{z}(t))\|_2 \le \int_0^h LMu\,du = \tfrac{1}{2}LMh^2.$$

Moreover,

$$\|S_h(\mathbf{z}(t)) - \mathbf{z}(t) - hG(\mathbf{z}(t))\|_2 \le h^2L_2M_1.$$

Thus, with $e_k = \|\mathbf{z}^k - \mathbf{z}(kh)\|_2$, one has $e_{k+1} \le e^{Lh}e_k + Ch^2$ and $e_0 = 0$. Summation yields

$$e_K \le Ch^2\frac{e^{LT} - 1}{e^{Lh} - 1} \le Ch\frac{e^{LT} - 1}{L} = \frac{CT}{K}\frac{e^{LT} - 1}{L}. \tag{D.45}$$

The constants depend on the continuous trajectory at fixed T, not on K. Finally, apply the triangle inequality

$$\|\mathbf{X}^{(K)} - \mathbf{X}^*\|_F \le \|\mathbf{X}^{(K)} - \mathbf{X}(T)\|_F + \|\mathbf{X}(T) - \mathbf{X}_{\mathrm{eq}}\|_F + \|\mathbf{X}_{\mathrm{eq}} - \mathbf{X}^*\|_F$$

and combine Eq. (D.43)–Eq. (D.45). □

The three terms in Eq. (D.42) separate the effects of finite depth, finite evolution time, and learned-module inconsistency. For a fixed horizon T, increasing K reduces the discretization bound at rate $O(K^{-1})$ but does not reduce the other two terms. Increasing T reduces the transient term, while the discretization constant and bound may increase with T; the depth must therefore be chosen for the selected horizon. More precisely, for any $\eta > 0$, one can first choose a finite $T > 0$ with $e^{-(1-\rho)T}E_0 \le \eta/2$ and then choose K sufficiently large that the discretization term is at most $\eta/2$. This gives

$$\|\mathbf{X}^{(K)} - \mathbf{X}^*\|_F \le \eta + \frac{r}{1-r}(\varepsilon_1 + \varepsilon_2).$$

This bound supports using the network state as an approximate solution for subsequent refinement. It applies to the specified sequential updates and their state before readout; an unconstrained learned readout or unrestricted layer parameters require additional analysis.

E EXPERIMENTAL RESULTS

E.1 IMPLEMENTATION SETTING

Data generation. We generate 10,000 AVE instances of dimension $n = 1000$ and split them into training and validation sets in a $9:1$ ratio. Following (Yu et al., 2022; Ke & Ma, 2017; Ke, 2020), we construct the coefficient matrix using a singular value decomposition,

$$\mathbf{A} = \mathbf{P}_1 \mathbf{\Lambda} \mathbf{P}_2^\top, \tag{E.1}$$

where $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the singular values of $\mathbf{A}$. Each singular value is sampled from $U(1.01, 2)$, ensuring $\|\mathbf{A}^{-1}\|_2 < 1$. The components of the ground-truth solution $\mathbf{x}_i^*$ are sampled in $U(4, 6)$ with 50% probability to be positive or negative. We then set $\mathbf{b} = \mathbf{A}\mathbf{x}^* - |\mathbf{x}^*|$. Test instances are generated using the same procedure at the dimensions required by each experiment. The orthogonal factors used for testing, denoted by $\mathbf{P}_1'$ and $\mathbf{P}_2'$, are different from the training factors $\mathbf{P}_1$ and $\mathbf{P}_2$, including when the training and testing dimensions coincide. In cross-size experiments, the test factors have the dimensions of the corresponding test problems and therefore cannot be shared with the training factors.

Network hyperparameters and training. We set the number of MFPI-Net layers to $K = 10$, the channel expansion dimension to $d = 16$, and the number of subchannels to $C_\varphi = 64$, using the notation in Eq. (7). Both $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$ have hidden dimension $d_{\mathbf{Q}} = 16$ and $K_{\mathbf{Q}} = 3$ layers. All parameters of MFPI-Net are initialized from the standard normal distribution $\mathcal{N}(0, 1)$. MFPI-Net is trained on the dataset with problem dimension $n = 1000$ for 1000 epochs using a learning rate of 10^{-3}.

Testing configuration. Table 4 reports the testing configuration. All methods within each comparison use the same batch size; batch sizes may differ across problem dimensions, as listed in the table. To further assess generalization to larger problems, we generate twenty test instances for each of the dimensions $n = 15000, 20000$, and 25000. For the experiments beyond the sufficient condition for unique solvability in Section 4.2, the training and testing settings are identical to those in Section 4.1, except that the singular values of $\mathbf{A}$ are sampled from $U(0.8, 2.0)$.

Table 4: Configuration of the testing dataset.

Problem dimension n	Number of instances	Batch size
1000	1000	100
2000	1000	100
5000	100	10
10000	50	5

Baselines and evaluation metric. We compare MFPI-Net with MFPI (Yu et al., 2022) and SOR-like methods (Ke & Ma, 2017; Chen et al., 2024). MFPI-Net and all AVE baselines are implemented in PyTorch and executed on the GPU using single-precision floating-point tensors (`torch.float32`) throughout training and evaluation. Whenever an explicit matrix inverse is required, it is computed on the GPU using `torch.linalg.inv` in the same precision. The residual error is defined as

$$\mathrm{RES}(\mathbf{A}\mathbf{x} - |\mathbf{x}|, \mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{A}_i\mathbf{x} - |x_i| - b_i)^2, \tag{E.2}$$

where $\mathbf{A}_i$ denotes the i-th row of $\mathbf{A}$.

Timing protocol. All reported solution times are average per-instance end-to-end solving times. Input coefficients $(\mathbf{A}, \mathbf{b})$ are transferred to the GPU before timing begins, so CPU-to-GPU data transfer is excluded. We call `torch.cuda.synchronize()` immediately before starting the timer and after the solver finishes, before stopping the timer, to ensure that all GPU operations within the timed interval have completed. For batched evaluation, the total elapsed solving time

across the evaluated batches is divided by the total number of instances. All methods within each comparison use the same batch size.

The timed interval starts from the original input coefficients already resident on the GPU and ends when the method returns its computed solution at the accuracy reported in the corresponding experiment. It includes all computational steps required for solving, including matrix preparation, parameter selection, explicit matrix inversion, and iterative updates or neural inference. An inverse is computed once and reused across subsequent iterations whenever its underlying matrix remains unchanged; its initial computation is included in the reported time. For two-stage methods and neural warm-start approaches, the reported time includes both the initial prediction and the subsequent numerical refinement.

Computing environment. All experiments are conducted on a computing platform equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM) and a 72-core Intel(R) Xeon(R) Gold 6139M CPU @ 2.30 GHz. The software dependencies include Python 3.8, PyTorch 2.2.2, OR-Tools 9.12.4544, and NumPy 1.23.2.

E.2 Numerical results on generalization and robustness evaluations

We investigate the generalization ability of the proposed method beyond the training set, with an emphasis on its adaptability to larger problems. To this end, we train MFPI-Net on a dataset with dimension $n = 1000$ and test it on large-scale AVE problems with dimensions ranging from 1,000 to 25,000. Table 5 reports the performance across these increasing dimensions, where SA denotes the accuracy of the signs predicted by MFPI-Net for each element of $\mathbf{x}^*$. As shown in Table 5, the residual error remains on the order of 10^{-5} across the tested problem scales. Furthermore, the model achieves 100% sign accuracy across all tested dimensions. These results corroborate the model's promising scalable generalization ability and robustness across varying problem scales.

Table 5: Generalization performance of MFPI-Net across increased problem dimensions.

	n				
	1000	10000	15000	20000	25000
Time	0.0037	0.409	0.549	1.009	1.801
RES	8.11e-5	7.92e-5	7.40e-5	7.45e-5	7.94e-5
SA	1.0	1.0	1.0	1.0	1.0

E.3 Numerical results on ablation study

To evaluate the effectiveness of the paired neural modules, the reconstruction loss, and the learnable smoothing of the absolute-value operator, we conduct a comprehensive ablation study across various problem scales $n \in \{1000, 5000, 8000, 10000\}$. Besides the original MFPI-Net, we consider four variants:

- Without regularization: The network architecture is maintained, but the model is trained without the auxiliary reconstruction loss term in Eq. (11).
- Scalar $q, 1/q$: The functions $f_{\mathbf{Q}}$ and $f_{\mathbf{Q}^{-1}}$ are replaced with scalar parameters q and $1/q$, respectively. This is a scalar linear-module variant; the remaining learned updates and activations are retained, so it is not the original MFPI algorithm.
- No $f_{\mathbf{Q}}, f_{\mathbf{Q}^{-1}}$: Both functional modules are removed entirely from the network.
- Without smoothing (exact $|\cdot|$): The learnable surrogate $\varphi_k(\mathbf{X}) \odot \mathbf{X}$ is replaced with the element-wise absolute-value operator $|\mathbf{X}|$, while the other network components are retained.

We perform five independent trials for each configuration and report the average result. All other experimental settings follow Section 4.1 and Appendix E.1.

Table 6 reports the residual error (RES), sign accuracy (SA), and end-to-end solution time for each configuration. Removing the reconstruction loss or simplifying the paired neural modules increases

Table 6: Experimental results of ablation study.

Method		n			
		1000	5000	8000	10000
MFPI-Net	Time	0.0037	0.105	0.279	0.409
	RES	8.110×10^{-5}	7.680×10^{-5}	8.075×10^{-5}	7.918×10^{-5}
	SA	1.0	1.0	1.0	1.0
without reconstruction loss	Time	0.0028	0.108	0.281	0.431
	RES	1.10×10^{-3}	1.07×10^{-3}	1.12×10^{-3}	1.11×10^{-3}
	SA	1.0	0.999	0.998	0.999
scalar $q, 1/q$	Time	0.0034	0.105	0.280	0.440
	RES	1.36×10^{-3}	1.34×10^{-3}	1.33×10^{-3}	1.31×10^{-3}
	SA	0.999	0.999	0.998	0.999
no $f_{\mathbf{Q}}, f_{\mathbf{Q}^{-1}}$	Time	0.0034	0.103	0.282	0.434
	RES	1.24×10^{-3}	1.13×10^{-3}	1.29×10^{-3}	1.15×10^{-3}
	SA	0.999	0.998	0.999	0.993
exact $\lvert\cdot\rvert$	Time	0.0033	0.099	0.246	0.407
	RES	0.0273	0.0305	0.0356	0.0492
	SA	0.998	0.998	0.996	0.995

RES from approximately 8×10^{-5} to 10^{-3} and reduces sign accuracy in most tested cases. These results support the contribution of both the paired modules and inverse-consistency regularization within the tested architecture and training setup. They do not establish a benefit from the classical choice of $\mathbf{Q}$ or an inherent expressiveness limitation of FPI unrolling; replacing the modules also changes model capacity.

Replacing the learnable surrogate $\varphi_k(\mathbf{X}) \odot \mathbf{X}$ with the exact absolute-value operator leads to the largest accuracy degradation among the variants: RES ranges from 0.0273 to 0.0492, compared with 7.680×10^{-5} to 8.110×10^{-5} for the full model. Sign accuracy also decreases to 99.5%–99.8%, whereas the full model achieves 100% at every tested dimension. Thus, the learnable smoothing contributes to both the quality of the approximate solution and the sign predictions used by the two-stage framework.

The runtime differences do not offset this loss of accuracy. The exact-absolute-value variant is slightly faster at $n = 5000, 8000$, and 10000, but slower at $n = 1000$ (0.0153 versus 0.0037 seconds). The other variants have broadly similar solution times to the full model. Overall, the full MFPI-Net achieves the lowest residual error and the highest sign accuracy across all tested dimensions.

E.4 Numerical results on different data distributions

To evaluate performance across a broader range of data distributions, we consider three additional configurations based on the data generation procedure in Appendix E.1:

1. **Varying spectral properties:** the singular values of $\mathbf{A}$ are sampled from $1 + |Z|$, where $Z \sim \mathcal{N}(0, 1)$.
2. **Ill-conditioned matrices:** one singular value is fixed at 100, while the remaining singular values are sampled from $U(1.01, 2)$, yielding a condition number $\kappa_2(\mathbf{A}) \approx 100$.
3. **Asymmetric solution structures:** the magnitudes of the components of $\mathbf{x}^*$ are sampled from $U(1, 10)$, with 90% positive entries and 10% negative entries.

Tables 7, 8, and 9 report the corresponding results for dimensions $n \in \{1000, 2000, 5000, 10000\}$. Here, SA denotes the accuracy of the sign patterns predicted by MFPI-Net. In all three configurations, MFPI-Net achieves 100% sign accuracy across the tested dimensions, and the two-stage method attains residual errors on the order of 10^{-11} to 10^{-9}. The two-stage method is faster than

both MFPI and SORLaopt at every tested dimension. These results support the effectiveness of the two-stage framework across the tested spectral and solution distributions.

Table 7: Numerical results for AVE instances with varying spectral properties.

Method		n			
		1000	2000	5000	10000
Two-stage MFPI-Net	Time	**0.011**	**0.032**	**0.275**	**1.154**
	RES	3.76e-11	1.07e-11	1.03e-10	1.93e-10
	SA	1.0	1.0	1.0	1.0
MFPI	Time	0.023	0.063	0.531	3.652
	RES	8.41e-9	9.11e-9	9.04e-9	9.58e-9
	Iters	69.0	68.0	67.0	72.0
SORLaopt	Time	0.018	0.066	0.545	2.726
	RES	9.23e-9	9.12e-9	9.33e-9	9.83e-9
	Iters	110.0	110.0	109.0	108.5

Table 8: Numerical results for AVE instances with ill-conditioned coefficient matrices.

Method		n			
		1000	2000	5000	10000
Two-stage MFPI-Net	Time	**0.011**	**0.032**	**0.273**	**1.153**
	RES	3.14e-9	1.41e-9	1.51e-10	1.99e-10
	SA	1.0	1.0	1.0	1.0
MFPI	Time	0.023	0.065	0.554	2.724
	RES	9.48e-10	8.87e-9	8.59e-9	9.00e-9
	Iters	82.0	81.0	80.0	79.0
SORLaopt	Time	0.019	0.067	0.556	2.774
	RES	9.46e-9	9.86e-9	9.60e-9	9.61e-9
	Iters	127.0	125.0	124.0	125.5

Table 9: Numerical results for AVE instances with asymmetric solution structures.

Method		n			
		1000	2000	5000	10000
Two-stage MFPI-Net	Time	**0.011**	**0.032**	**0.275**	**1.200**
	RES	4.67e-11	1.60e-10	1.16e-10	2.20e-10
	SA	1.0	1.0	1.0	1.0
MFPI	Time	0.026	0.077	0.615	3.045
	RES	9.81e-9	9.98e-9	9.69e-9	9.79e-9
	Iters	361.0	354.0	348.0	354.0
SORLaopt	Time	0.017	0.061	0.513	2.654
	RES	8.94e-9	7.82e-9	9.85e-9	8.83e-9
	Iters	65.0	65.0	64.0	64.5

E.5 Sign prediction for solution components near zero

To assess sign prediction when solution components can be close to zero, we sample their magnitudes from $U(0.001, 1)$, with half of the entries positive and half negative. Small-magnitude components make sign identification more sensitive to prediction errors: even a small error in the approximate solution can change the predicted sign. We evaluate MFPI-Net and its two-stage extension at dimensions $n \in \{1000, 2000, 5000, 10000\}$.

Table 10 reports the residual error (RES) and component-wise sign accuracy (SA) of the neural predictions, together with the end-to-end solution times, average second-stage iteration counts, and

final residual errors of the two-stage method. For two-stage MFPI-Net, Iters counts only the second-stage iterations, initialized with the MFPI-Net output.

Table 10: Sign prediction and two-stage solving for AVEs with solution components near zero.

Method		n			
		1000	2000	5000	10000
MFPI-Net	RES	6.08e-5	9.37e-5	1.22e-4	2.65e-4
	SA	0.999	0.998	0.998	0.998
Two-stage MFPI-Net	Time	0.026	0.041	0.386	1.673
	Iters	1.8	1.95	2.0	2.0
	RES	2.05e-11	8.47e-11	4.43e-11	2.93e-11

MFPI-Net achieves high component-wise sign accuracy, ranging from 99.8% to 99.9%. Nevertheless, relatively few instances have every component sign predicted correctly. Component-wise accuracy and exact recovery of an entire sign pattern are distinct: a small fraction of incorrect signs can still leave multiple errors in a high-dimensional instance. Thus, high SA alone does not ensure that a single second-stage iteration recovers the solution.

Despite these remaining sign errors, the two-stage method obtains high-accuracy solutions using only 1.8–2.0 second-stage iterations on average. Across the tested dimensions, the final residual errors range from 2.05×10^{-11} to 8.47×10^{-11}, with end-to-end solution times ranging from 0.026 to 1.673 seconds.

These results show that exact recovery of the entire sign pattern is not necessary for the two-stage framework to be effective on the tested instances. The high sign accuracy of MFPI-Net provides a useful warm start, allowing the second-stage iterations to correct the remaining sign errors and reach a high-accuracy solution in a small number of steps.

E.6 Failure analysis of classical solvers beyond the standard convergence regime

Section 4.1 primarily compares our method with FPI-family solvers. To broaden this comparison, Section 4.2 additionally considers generalized Newton (Mangasarian, 2009). On the relatively easy instances in Section 4.1, generalized Newton converges rapidly, leaving limited scope to assess the benefit of our learned solver. We therefore construct more challenging instances by keeping the same matrix construction and shifting the singular-value sampling interval from $U(1.01, 2)$ to $U(0.8, 2.0)$. The resulting instances no longer satisfy $\|\mathbf{A}^{-1}\|_2 < 1$ and lie outside the standard guaranteed convergence regime of the FPI-family methods.

Table 11 reports residuals of basic FPI and generalized Newton at increasing iteration counts for $n \in \{1000, 5000, 10000\}$ under the setting of Section 4.2. Both methods fail to converge on these tested instances; the residuals document solver failure, and their precise magnitudes are not meaningful measures of relative performance.

Table 11: Failure of FPI and generalized Newton on AVEs with $\|\mathbf{A}^{-1}\|_2 \geq 1$: residuals at increasing iteration counts.

n	Method	Iteration count				
		0	1000	2000	5000	10000
1000	Generalized Newton	25.3	44278616	48442588	1418044800	inf
	FPI	25.3	25.55	26.43	3.8×10^{16}	inf
5000	Generalized Newton	25.3	49376293	78246285	1272478945	inf
	FPI	25.3	30.87	4.96×10^{8}	5.3×10^{15}	inf
10000	Generalized Newton	25.3	37291504	81502649	1259471036	inf
	FPI	25.3	27.21	1.53×10^{5}	6.8×10^{13}	inf

inf denotes a non-finite residual returned by the computation.

For FPI, the interval shift removes the contraction guarantee, and basic FPI diverges in these experiments; the prescribed MFPI and SOR-like parameter-selection rules likewise yield no admissible parameters.

For generalized Newton, the baseline initial points lie outside the solution's basin of attraction, leading to failure on the tested instances. In contrast, MFPI-Net predicts the correct solution sign pattern and thus supplies an initial point within this basin: when $\mathrm{sgn}(\mathbf{x}^{\mathrm{NN}}) = \mathrm{sgn}(\mathbf{x}^*)$ and $\mathbf{A} - \mathbf{D}(\mathbf{x}^*)$ is nonsingular, one Newton step in Eq. (22) recovers $\mathbf{x}^*$. This explains why Newton refinement succeeds from the learned initialization, yielding the small residuals reported in Table 2.

E.7 Numerical results on PageRank-derived AVE instances

On the tested PageRank-derived AVE instances, the generalized Newton method (Mangasarian, 2009) cannot proceed with its iteration because the linear system that must be solved to compute the next iterate is inconsistent, i.e., it has no solution. Table 12 therefore reports the results of MFPI-Net and FPI for dimensions $n \in \{200, 500, 1000\}$. Increasing the FPI iteration count from 10,000 to 30,000 yields only limited reductions in residual error, with residuals remaining between 0.0091 and 0.0157 after 30,000 iterations. In contrast, MFPI-Net produces approximate solutions with residuals on the order of 10^{-6} and requires less solution time than FPI at every reported iteration budget and dimension.

Table 12: Numerical results on PageRank-derived AVE instances.

Size		FPI			MFPI-Net
		10000	20000	30000	
200	Time	0.049	0.077	0.110	0.0031
	RES	0.0170	0.0161	0.0157	2.73e-6
500	Time	0.163	0.313	0.463	0.0047
	RES	0.0123	0.0115	0.0110	4.13e-6
1000	Time	0.622	1.208	1.788	0.0120
	RES	0.0104	0.0096	0.0091	2.02e-6

E.8 Comparison with other two-stage methods for solving LPs

We compare MFPI-Net+PDLP with PDHG-Net+PDLP (Li et al., 2024a), using cold-start PDLP as the baseline on PageRank-derived LPs of sizes $n \in \{200, 500, 1000, 5000, 10000, 20000\}$. Both two-stage methods supply a learned initialization to the same PDLP refinement solver, with a relative optimality tolerance of 10^{-4}. MFPI-Net obtains this initialization through the AVE reformulation, whereas PDHG-Net directly unrolls an LP-specific algorithm.

Table 13 shows that both two-stage methods reduce PDLP iterations and solving time relative to cold starts at all six sizes. At $n = 200, 500, 1000$, MFPI-Net+PDLP outperforms PDHG-Net+PDLP in both metrics: MFPI-Net reduces iterations by 64.3%–76.1% and solving time by 32.7%–50.7%, compared with 47.7%–57.2% and 23.4%–38.1%, respectively, for PDHG-Net. These results demonstrate that learned solutions of the AVE reformulation can provide effective LP initializations even compared with an LP-specific learned warm-start method.

At $n = 5000, 10000, 20000$, PDHG-Net+PDLP achieves larger reductions in both metrics, while MFPI-Net still reduces iterations by 32.0%–44.5% and solving time by 26.9%–36.5%. This advantage is consistent with PDHG-Net's algorithmic structure: PDLP is based on PDHG (Applegate et al., 2021), and PDHG-Net unrolls PDHG (Li et al., 2024a), making it naturally aligned with the refinement solver. Moreover, PDHG as used in PDLP targets large-scale LPs, so an unfolded network based on this algorithm is well suited to this setting. MFPI-Net has a broader scope: it applies not only to LPs but also to other problems admitting AVE reformulations, including the convex quadratic programs (CQPs) evaluated in Appendix E.9. Finally, MFPI-Net is trained at $n = 1000$ for these large-scale tests, which may limit its performance.

Table 13: Comparison with other two-stage methods for solving LPs.

Size		PDLP	MFPI-Net+ PDLP	PDHG-Net+ PDLP
200	Time	0.107	0.072 (↓ 32.7%)	0.082 (↓ 23.4%)
	Iters	611.20	**218.24** (↓ 64.3%)	315 (↓ 48.4%)
500	Time	0.677	0.380 (↓ 43.8%)	0.479 (↓ 29.1%)
	Iters	651.52	**184.32** (↓ 71.7%)	340.5 (↓ 47.7%)
1000	Time	3.266	1.611 (↓ 50.7%)	2.019 (↓ 38.1%)
	Iters	751.36	**179.20** (↓ 76.1%)	321.3 (↓ 57.2%)
5000	Time	117.98	74.88 (↓ 36.5%)	59.79 (↓ 49.3%)
	Iters	1408.0	780.8 (↓ 44.5%)	**713.32** (↓ 49.3%)
10000	Time	636.31	419.66 (↓ 34.0%)	342.46 (↓ 46.2%)
	Iters	1753.6	1088.2 (↓ 37.9%)	**943.7** (↓ 46.2%)
20000	Time	3050.04	2226.89 (↓ 26.9%)	1739.12 (↓ 42.9%)
	Iters	1905.9	1296.0 (↓ 32.0%)	**1086.7** (↓ 42.9%)

E.9 Numerical results on QP-derived AVE instances

We further evaluate the performance of MFPI-Net on AVE instances derived from the representative RHS problems (Xiong et al., 2025; Donti et al., 2021), which are formulated as follows:

$$\min_{\mathbf{x}_p} \ \frac{1}{2}\mathbf{x}_p^\top \mathbf{P}\mathbf{x}_p + \mathbf{c}^\top \mathbf{x}_p, \qquad \text{s.t. } \mathbf{G}_1\mathbf{x}_p = \mathbf{h}_1, \quad \mathbf{G}_2\mathbf{x}_p \le \mathbf{h}_2, \tag{E.3}$$

where $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with entries sampled from a uniform distribution $U(0,1)$. The components of vectors $\mathbf{c}, \mathbf{h}_1$, and $\mathbf{h}_2$ are similarly drawn from $U(0,1)$, while the elements of the constraints matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are independent and identically distributed (i.i.d) random variables following a standard normal distribution $\mathcal{N}(0,1)$. The number of both equality and inequality constraints is set to $n/4$. Experiments are conducted on these RHS-derived instances across dimensions $n \in \{100, 200, 400\}$, and the results are summarized in Table 14.

Table 14: Numerical results on RHS derived AVE instances.

Size		FPI			MFPI-Net
		10000	20000	30000	
100	Time	0.043	0.070	0.095	0.0052
	RES	5.103	2.006	0.0097	0.0040
200	Time	0.064	0.117	0.164	0.0087
	RES	5.658	5.547	5.499	0.0070
400	Time	0.191	0.368	0.545	0.0172
	RES	1.909	1.828	1.814	0.0062

On these RHS-CQP instances, MFPI-Net achieves lower residual errors and shorter solution times than FPI at every reported iteration budget and problem size. After $30{,}000$ FPI iterations, the residual errors are 0.0097, 5.499, and 1.814 for $n = 100, 200$, and 400, respectively, compared with 0.0040, 0.0070, and 0.0062 for MFPI-Net. To assess the practical utility of these approximate solutions, we use them to warm-start the splitting conic solver (SCS) for solving Eq. (E.3). As summarized in Table 15, MFPI-Net warm starts reduce SCS iteration counts at all three problem sizes. Total solution time decreases for $n = 200$ and $n = 400$, whereas it increases for $n = 100$ despite the lower iteration count.

Table 15: Performance comparison of SCS initialized with warm-starts from different AVE solvers.

size		SCS	FPI SCS	MFPI-Net SCS
100	Time	0.076	0.165 (↑117%)	0.105 (↑38.2%)
	Iters	1214.5	514.0 (↓57.7%)	591.75 (↓51.3%)
200	Time	0.507	0.792 (↑56.1%)	0.392 (↓22.7%)
	Iters	2965.25	3165.5 (↑6.75%)	1610.0 (↓45.7%)
400	Time	3.092	3.836 (↑24.0%)	2.445 (↓20.9%)
	Iters	4748.75	5097.0 (↑7.33%)	3586.5 (↓24.5%)