Understanding Deep Learning with Reasoning Layer

Anonymous Authors[1]

Abstract

Recently, there is a surge of interest in combining deep learning models with reasoning in order to handle more sophisticated learning tasks. In many cases, a reasoning task can be solved by an iterative algorithm. This algorithm is often unrolled, and used as a specialized layer in the deep architecture, which can be trained end-to-end with other neural components. Although such hybrid deep architectures have led to many empirical successes, theoretical understandings of such architectures, especially the interplay between algorithm layers and other neural layers, remains largely unexplored. In this paper, we take an initial step toward an understanding of such hybrid deep architectures by showing that properties of the algorithm layers, such as convergence, stability and sensitivity, are intimately related to the approximation and generalization abilities of the end-to-end model. Furthermore, our analysis matches nicely with experimental observations under various conditions, suggesting that our theory can provide useful guidelines for designing deep architectures with reasoning layers.

1. Introduction

Many real world applications require perception and reasoning to work together to solve a problem. Perception refers to the ability to understand and represent inputs, while reasoning refers to the ability to follow prescribed steps and derive answers satisfying certain structures or constraints. To tackle such sophisticated learning tasks, recently, there is a surge of interests in combining deep perception models with reasoning modules.

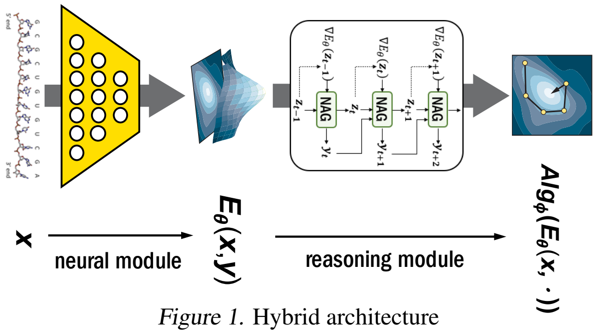

Figure 1. Hybrid architecture

Typically, a **reasoning module** is stacked on top of a **neural module**, and treated as an additional layer of the overall deep architecture; then all the parameters in the architecture are optimized end-to-end with loss gradients (Fig 1). Very often these reasoning modules can be implemented as unrolled *iterative algorithms*, which can solve more sophisticated tasks with carefully designed and interpretable operations. For instance, SATNet [1] integrated a satisfiability solver into its deep model as a reasoning module; E2Efold [2] used a constrained optimization algorithm on top of a neural energy network to predict and reasoning about RNA structures. [3] used optimal transport algorithm as a reasoning module for learning to sort. Other algorithms such as ADMM [4, 5], Langevin dynamics [6], inductive logic programming [7], DP [8], k-means clustering [9], belief propagation [10], power iterations [11] are also used as differentiable reasoning modules in deep models for various learning tasks. Thus in the reminder of the paper, we will use reasoning layer and algorithm layer interchangeably.

While these previous works have demonstrated the effectiveness of combining deep learning with reasoning, theoretical understandings of such hybrid deep architectures remain largely unexplored. For instance, what is the benefit of using a reasoning module based on unrolled algorithms compared to generic architectures such as RNN? How exactly will the reasoning module affect the generalization ability of the deep architecture? For different algorithms which can solve the same task, what are their differences when used as reasoning modules in deep models? Despite the rich literature on rigorous analysis of algorithm properties, there is a paucity of work leveraging these analyses to formally study the learning behavior of deep architectures containing algorithm layers. This motivates us to ask the intriguing and timely question of

> *How will the algorithm properties of a reasoning layer affect the learning behavior of deep archi-*

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tectures containing such layers?

In this paper, we provide a first step toward an answer to this question by analyzing the approximation and generalization abilities of such hybrid deep architectures. To the best our knowledge, such analysis has not been done before and is challenging in the sense that: 1) The analysis of certain algorithm properties such as convergence can be complex by itself; 2) Models based on highly structured iterative algorithms have rarely been analyzed before; 3) The bound needs to be sharp enough to match empirical observations. In this new setting, the complexity of algorithm analysis and generalization analysis intertwined together, making the analysis even more challenging.

Summary of results. We find that standard Rademacher complexity analysis, widely used for neural networks [12, 13, 14], becomes insufficient for explaining behaviors of hybrid architectures. Thus we resort to a more refined local Rademacher complexity analysis [15, 16], and find that:

- **Relation to algorithm properties.** Algorithm properties such as convergence, stability and sensitivity all play important roles in generalization ability of the hybrid architecture. Generally speaking, an algorithm layer that is faster converging, more stable and less sensitive will be able to better approximate the joint perception and reasoning task, while at the same time generalize better.
- **Which algorithm?** The tradeoff is that a faster converging algorithm has to be less stable [17]. Therefore, depending on the actual scenarios, the choice of a better algorithm layer can be different. Our theorem reveals that when the neural module is over- or under-parameterized, stability of the algorithm layer can be more important than its convergence; but when the neural module is about-the-right-parameterized, a faster converging algorithm layer may give a better generalization.
- **What depth?** With deeper algorithm layers, the representation ability gets better, but the generalization becomes worse if the neural module is over/under-parameterized. Only when it has about-the-right complexity, deeper algorithm layers can induce both better representation and generalization.
- **What if RNN?** It has been shown that RNN/GNN can also represent reasoning and iterative algorithms [18, 14]. We use RNN as an example in Appendix B to demonstrate that these generic reasoning modules can also be analyzed under our framework, which explains that RNN layers induce a better representation power but a worse generalization ability compared to traditional algorithm layers.
- **Experiments.** We conduct empirical experiments to validate our theory and show that it matches nicely with experimental observations under various conditions. These results suggest that our theory can provide useful practical guidelines for designing deep architectures with reasoning layers. Experimental results are presented in Appendix C.

Contributions and limitations. To the best of our knowledge, this is the first result to quantitatively characterize the effects of algorithm properties on the learning behavior of hybrid deep architectures with reasoning layers. Our result reveals an intriguing and previously unknown interplay and tradeoff between algorithm convergence, stability and sensitivity on the model generalization, and thus provides design principles for deep architectures with reasoning layers. To simplify analysis, our initial study is limited to a setting where the reasoning module is an unconstrained optimization algorithm and the neural module outputs a quadratic energy function. However, our analysis framework can be extended to more complicated case and the insights will apply beyond our current setting.

Related theoretical works. Our analysis borrows proof techniques for analyzing algorithm properties from the optimization literature [17, 19] and for bounding Rademacher complexity from the statistical learning literature [12, 15, 16, 20, 21], but our focus and results are new. More precisely, the ‘leave-one-out’ stability of optimization algorithms has been used to derive generalization bounds [22, 23, 24, 17, 25, 26]. However, all existing analyses are in the context where the optimization algorithms are used to train and select the model, while our analysis is based on a fundamentally different viewpoint where the algorithm itself is unrolled and integrated as a layer in the deep model. Also, existing works on the generalization of deep learning mainly focus on generic neural architectures such as feed-forward neural network, recurrent neural network, graph neural network, etc [12, 13, 14]. Complexity of models based on highly structured iterative algorithms and the relation to algorithm properties have not been investigated. Furthermore, we are not aware of previous use of local Rademacher complexity analysis in this context.

2. Setting: Optimization Algorithms as Reasoning Modules

Very often reasoning can be accomplished by solving an optimization problem defined by a neural perceptual module. For instance, visual SUDOKU puzzle can be solved using a neural module to perceive the digits and then using a quadratic optimization module to maximize a logic satisfiability objective [1]. RNA folding problem can be tackled using a neural energy model to capture pairwise relations between RNA bases and a constrained optimization module to minimize the energy with additional pairing constraints to obtain a folding [2]. In a broader context, MAML [27, 28] also has a neural module for joint initialization and a reasoning module that performs optimization steps for task-specific adaptation. Other examples include [29, 6, 30, 31, 32, 33, 34].

As an initial attempt to analyze deep architectures with reasoning layers, we will restrict our analysis to a simple case where $E_\theta(\boldsymbol{x}, \boldsymbol{y})$ in Fig.1 is quadratic in $\boldsymbol{y}$. A reason is that the analysis of advanced algorithms such as Nesterov accelerated gradients will become very complex for general cases. Similar problems occur in [17] which also restricts the proof to quadratic objectives. Specifically:

Problem Setting: Consider a hybrid architecture where the neural module is an energy function in form of $E_\theta((\boldsymbol{x}, \boldsymbol{b}), \boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^\top Q_\theta(\boldsymbol{x})\boldsymbol{y} + \boldsymbol{b}^\top \boldsymbol{y}$, where Q_θ is a neural network that maps $\boldsymbol{x}$ to a matrix. Each energy can be uniquely represented by $(Q_\theta(\boldsymbol{x}), \boldsymbol{b})$, so we can write the overall architecture as

$$f_{\phi,\theta}(\boldsymbol{x}, \boldsymbol{b}) := \texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}). \tag{1}$$

Given samples $S_n = \{((\boldsymbol{x}_1, \boldsymbol{b}_1), \boldsymbol{y}_1^*), \cdots, ((\boldsymbol{x}_n, \boldsymbol{b}_n), \boldsymbol{y}_n^*)\}$, where the labels $\boldsymbol{y}^*$ are given by the *exact minimizer* $\texttt{Opt}$ of the corresponding Q^*, i.e., $\boldsymbol{y}^* = \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})$, the learning problem is to find the best model $f_{\phi,\theta}$ from the space $\mathcal{F} := \{f_{\phi,\theta} : (\phi, \theta) \in \Phi \times \Theta\}$ by minimizing the empirical loss function

$$\min\nolimits_{f_{\phi,\theta}\in\mathcal{F}} \quad \tfrac{1}{n}\textstyle\sum_{i=1}^n \ell_{\phi,\theta}(\boldsymbol{x}_i, \boldsymbol{b}_i), \quad \text{where} \tag{2}$$

$\ell_{\phi,\theta}(\boldsymbol{x}, \boldsymbol{b}) := \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})\|_2$. Furthermore, we assume:

- Both Q_θ and Q^* map $\mathcal{X}$ to $\mathcal{S}_{\mu,L}^{d\times d}$ where $\mathcal{S}_{\mu,L}^{d\times d}$ is the space of symmetric positive definite (SPD) matrices with μ and L as its smallest and largest singular values. Thus the induced energy function E_θ will be μ-strongly convex and L-smooth, and the output of $\texttt{Opt}$ is unique.
- The input $(\boldsymbol{x}, \boldsymbol{b})$ is a pair of random variables where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^m$ and $\boldsymbol{b} \in \mathcal{B} \subseteq \mathbb{R}^d$. Assume $\boldsymbol{b}$ has mean $\mathbb{E}[\boldsymbol{b}] = \boldsymbol{0}$ and variance $\Sigma_{\boldsymbol{b}} = \sigma_b^2 I$. Assume $\boldsymbol{x}$ and $\boldsymbol{b}$ are independent, and their joint distribution follows a probability measure P. Assume samples in S_n are drawn i.i.d. from P.
- Assume $\mathcal{B}$ is bounded, and let $M = \sup_{(Q,\boldsymbol{b})\in\mathcal{S}_{\mu,L}^{d\times d}\times\mathcal{B}} \|\texttt{Opt}(Q, \boldsymbol{b})\|_2$.

Though this setting does not encompass the full complexity of hybrid deep architectures, it already reveals interesting connections between algorithm properties of the reasoning module and the learning behaviors of hybrid architectures.

3. Properties of Algorithms

In this section, we formally define the algorithm properties of the reasoning module $\texttt{Alg}_\phi^k$, under the problem setting presented in Sec 2. After that, we compare the corresponding properties of gradient descent, $\texttt{GD}_\phi^k$, and Nesterov's accelerated gradients, $\texttt{NAG}_\phi^k$, as concrete examples.

(I) Convergence rate of an algorithm portrays how fast the optimization error decreases as k grows. Formally, we say $\texttt{Alg}_\phi^k$ has a convergence rate *Cvg*(k, ϕ) if for any $Q \in \mathcal{S}_{\mu,L}^{d\times d}, \boldsymbol{b} \in \mathcal{B}$, $\|\texttt{Alg}_\phi^k(Q, \boldsymbol{b}) - \texttt{Opt}(Q, \boldsymbol{b})\|_2 \leq$ *Cvg*$(k, \phi)\|\texttt{Alg}_\phi^0(Q, \boldsymbol{b}) - \texttt{Opt}(Q, \boldsymbol{b})\|_2$.

(II) Stability of an algorithm characterizes its robustness to small *perturbations in the optimization objective*, which corresponds to the perturbation of Q and $\boldsymbol{b}$ in the quadratic case. For the purpose of this paper, we say an algorithm $\texttt{Alg}_\phi^k$ is *Stab*(k, ϕ)-stable if for any $Q, Q' \in \mathcal{S}_{\mu,L}^{d\times d}$ and $\boldsymbol{b}, \boldsymbol{b}' \in \mathcal{B}$, $\|\texttt{Alg}_\phi^k(Q, \boldsymbol{b}) - \texttt{Alg}_\phi^k(Q', \boldsymbol{b}')\|_2 \leq$ *Stab*$(k, \phi)\|Q - Q'\|_2 +$ *Stab*$(k, \phi)\|\boldsymbol{b} - \boldsymbol{b}'\|_2$, where $\|Q - Q'\|_2$ is the spectral norm of the matrix $Q - Q'$.

(III) Sensitivity characterizes the robustness to small *perturbations in the algorithm parameters* ϕ. We say the sensitivity of $\texttt{Alg}_\phi^k$ is *Sens*(k) if it holds for all $Q \in \mathcal{S}_{\mu,L}^{d\times d}, \boldsymbol{b} \in \mathcal{B}$, and $\phi, \phi' \in \Phi$ that $\|\texttt{Alg}_\phi^k(Q, \boldsymbol{b}) - \texttt{Alg}_{\phi'}^k(Q, \boldsymbol{b})\|_2 \leq$ *Sens*$(k)\|\phi - \phi'\|_2$. This concept is referred in the deep learning community to "parameter perturbation error" or "sharpness" [35, 36, 37]. It has been used for deriving generalization bounds of neural networks, both in the Rademacher complexity framework [12] and PAC-Bayes framework [38].

(IV) Stable region is the range Φ of the parameters ϕ where the algorithm output will remain bounded as k grows to infinity, i.e., numerically stable. Only when the algorithms operate in the stable region, the corresponding *Cvg*(k, ϕ), *Stab*(k, ϕ) and *Sens*(k) will remain finite for all k. It is usually very difficult to identity the exact stable region, but a sufficient range can be provided.

GD and NAG. Now we will compare the above four algorithm properties for gradient descent and Nesterov's accelerated gradient method, both of which can be used to solve the quadratic optimization in our problem setting. Let $\texttt{GD}_\phi$ and $\texttt{NAG}_\phi$ denote the algorithm update steps of GD and NAG, where the hyperparameter ϕ corresponds to the step size. Denote the results of k-step update of GD and NAG by $\texttt{GD}_\phi^k(Q, \boldsymbol{b})$ and $\texttt{NAG}_\phi^k(Q, \boldsymbol{b})$, respectively. The initializations in the algorithms are set to be zero vectors throughout this paper. Then their algorithm properties are summarized in Table 2 in Appendix D, which shows *(i) Convergence*: NAG converges faster than GD. *(ii) Stability*: However, as k grows, NAG is less stable than GD for a fixed k, in contrast to their convergence behaviors. This is pointed out in [17], which proves that a faster converging algorithm has to be less stable. *(iii) Sensitivity*: The sensitivity behaves similar to the convergence, where NAG is less sensitive to step-size perturbation than GD. Also, the sensitivity of both algorithms gets smaller as k grows larger. *(iv): Stable region*: The stable region of GD is larger than that of NAG. It means a larger step size is allowable for GD that will not lead to exploding outputs even if k is large. Note that all the other algorithm properties are based on the assumption that ϕ is in the stable region Φ. Furthermore, as $k \to \infty$, the space

Theorem 3.1. *Assume the problem setting in Sec 2. Then we have for any $t > 0$, it holds true that*

$$\mathbb{E}R_n\ell_{\mathcal{F}}^{loc}(r) \leq \sqrt{2}dn^{-\frac{1}{2}}\mathit{Stab}(k)\left(\sqrt{(\mathit{Cvg}(k)M+\sqrt{r})^2C_1(n)+C_2(n,t)}+C_3(n,t)+4\right)+\mathit{Sens}(k)B_\Phi, \tag{3}$$

where $B_\Phi = \frac{1}{2}\sup_{\phi,\phi'\in\Phi}\|\phi-\phi'\|_2$, $\mathit{Stab}(k) = \sup_\phi \mathit{Stab}(k,\phi)$, $\mathit{Cvg}(k) = \sup_\phi \mathit{Cvg}(k,\phi)$, and C_i are constants monotone in the covering number $\mathcal{N}(\frac{1}{\sqrt{n}}, \ell_\mathcal{Q}, L_\infty)$ of $\ell_\mathcal{Q}$ with radius $\frac{1}{\sqrt{n}}$ and L_∞ norm. We refer their exact definitions to Appendix F.

$\{\mathtt{Alg}_\phi^k : \phi \in \Phi\}$ will finally shrink to a single function, which is the exact minimizer $\{\mathtt{Opt}\}$.

How will the algorithm properties affect the learning behavior of deep architecture with reasoning layers? We provide the approximation ability analysis in Appendix A and the generalization analysis in the next section.

4. Generalization Ability

How will algorithm properties affect the generalization ability of deep architectures with reasoning layers? We are interested in the *generalization gap* between the expected loss and empirical loss, $P\ell_{\phi,\theta} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{b}}\ell_{\phi,\theta}(\boldsymbol{x},\boldsymbol{b})$ and $P_n\ell_{\phi,\theta} = \frac{1}{n}\sum_{i=1}^n \ell_{\phi,\theta}(\boldsymbol{x}_i,\boldsymbol{b}_i)$, respectively, where P_n is the empirical probability measure induced by the samples S_n. Let $\ell_\mathcal{F} := \{\ell_{\phi,\theta} : \phi \in \Phi, \theta \in \Theta\}$ be the function space of losses of the models. The generalization gap, $P\ell_{\phi,\theta} - P_n\ell_{\phi,\theta}$, can be bounded by the Rademacher complexity, $\mathbb{E}R_n\ell_\mathcal{F}$, which is defined as the expectation of the empirical Rademacher complexity, $R_n\ell_\mathcal{F} := \mathbb{E}_{\boldsymbol{\sigma}}\sup_{\phi\in\Phi,\theta\in\Theta}\frac{1}{n}\sum_{i=1}^n\sigma_i\ell_{\phi,\theta}(\boldsymbol{x}_i,\boldsymbol{b}_i)$, where $\{\sigma_i\}_{i=1}^n$ are n independent Rademacher random variables uniformly distributed over $\{\pm 1\}$. Generalization bounds derived from Rademacher complexity have been studied in many works [39, 40, 41].

Main Results: [Theorem 3.1]. More specifically, the local Rademacher complexity of $\ell_\mathcal{F}$ at level r is defined as $\mathbb{E}R_n\ell_\mathcal{F}^{loc}(r)$ where $\ell_\mathcal{F}^{loc}(r) := \{\ell_{\phi,\theta} : \phi \in \Phi, \theta \in \Theta, P\ell_{\phi,\theta}^2 \leq r\}$. This notion is less general than the one defined in [15, 16] but is sufficient for our purpose. Here we also define a losses function space $\ell_\mathcal{Q} := \{\|Q_\theta - Q^*\|_F : \theta \in \Theta\}$ for the neural module Q_θ. With these definitions, Theorem 3.1 shows that the local Rademacher complexity of the hybrid architecture is intimately related to all aspects of algorithm properties, namely convergence, stability and sensitivity, and there is an intriguing trade-off.

Trade-offs between convergence, stability and sensitivity. Generally speaking, the algorithm convergence $\mathit{Cvg}(k)$ and sensitivity $\mathit{Sens}(k)$ have similar behavior, but $\mathit{Stab}(k)$ behaves opposite to them. See illustrations in Fig 2. Therefore, the way these three quantities interplay in Theorem 3.1 introduces an intriguing trade-off among them, suggesting in different regime, one may see different generalization behavior. More specially, depending on the parameterization of Q_θ, the coefficients C_1, C_2, and C_3 in Eq. 3 may have different scale, making the local Rademacher complexity bound dominated by different algorithm properties. Since the coefficients C_i are monotonely increasing in the covering number of $\ell_\mathcal{Q}$, we expect that: **(i)** When Q_θ is **over**-parameterized, the covering number of $\ell_\mathcal{Q}$ becomes large, so as the three coefficients. Large C_i will reduce the effect of $\mathit{Cvg}(k)$ and make Eq. 3 dominated by $\mathit{Stab}(k)$; **(ii)** Inversely, when Q_θ is **under**-parameterized, the three coefficients get small, but they still reduce the effect of $\mathit{Cvg}(k)$ given the constant 4 in Eq. 3, again making it dominated by $\mathit{Stab}(k)$; **(iii)** When Q_θ has **about-the-right** parameterization, we can expect $\mathit{Cvg}(k)$ to play critical roles in Eq. 3 which will then behave similar to the product $\mathit{Stab}(k)\mathit{Cvg}(k)$, as illustrated schematically in Fig 2. We experimentally validate these implications in Sec C.

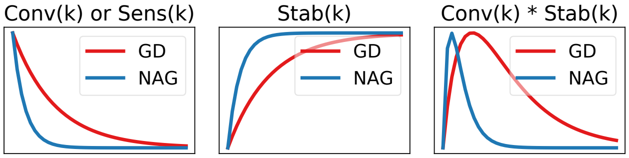

Figure 2. Overall trend of algorithm properties.

Trade-off of the depth. Combining the above implications with the approximation ability analysis in Sec A, we can see that in the above-mentioned cases **(i)** and **(ii)**, deeper algorithm layers will lead to better approximation accuracy but worse generalization. Only in the ideal case **(iii)**, a deeper reasoning module can induce both better representation and generalization abilities. This result provides practical guidelines for some recently proposed infinite-depth models [42, 43].

5. Conclusion and Discussion

In this paper, we take an initial step toward the theoretical understanding of deep architectures with reasoning layers. Our theorem indicates intriguing relation between algorithm properties of the reasoning module and the approximation and generalization of the hybrid architecture, which in turns provide practical guideline for designing reasoning layers. The assumptions we made in the problem setting are only for avoiding the non-uniqueness of the reasoning solution and the instability of the mapping from the reasoning solution to the neural module. The assumptions could be relaxed if we can involve other techniques to resolve these issues.

References

[1] Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pages 6545–6554, 2019.

[2] Xinshi Chen, Yu Li, Ramzan Umarov, Xin Gao, and Le Song. Rna secondary structure prediction by learning unrolled algorithms. *arXiv preprint arXiv:2002.05810*, 2020.

[3] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable sorting using optimal transport: The sinkhorn cdf and quantile operator. *arXiv preprint arXiv:1905.11885*, 2019.

[4] Harsh Shrivastava, Xinshi Chen, Binghong Chen, Guanghui Lan, Srinivas Aluru, Han Liu, and Le Song. GLAD: Learning sparse graph recovery. In *International Conference on Learning Representations*, 2020.

[5] Y Yang, J Sun, H Li, and Z Xu. Admm-net: A deep learning approach for compressive sensing mri. corr. *arXiv preprint arXiv:1705.06869*, 2017.

[6] John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations*, 2019.

[7] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, pages 3749–3759, 2018.

[8] Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *35th International Conference on Machine Learning*, volume 80, 2018.

[9] Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. End to end learning and optimization on graphs. In *Advances in Neural Information Processing Systems*, pages 4674–4685, 2019.

[10] Justin Domke. Parameter learning with truncated message-passing. In *CVPR 2011*, pages 2937–2943. IEEE, 2011.

[11] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. In *Advances in Neural Information Processing Systems*, pages 3156–3164, 2019.

[12] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[13] Minshuo Chen, Xingguo Li, and Tuo Zhao. On generalization bounds of a family of recurrent neural networks. *arXiv preprint arXiv:1910.12947*, 2019.

[14] Vikas K Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. *arXiv preprint arXiv:2002.06157*, 2020.

[15] Peter L Bartlett, Olivier Bousquet, Shahar Mendelson, et al. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[16] Vladimir Koltchinskii et al. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

[17] Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

[18] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[19] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[20] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

[21] Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Structured prediction theory based on factor graph complexity. In *Advances in Neural Information Processing Systems*, pages 2514–2522, 2016.

[22] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[23] Shivani Agarwal and Partha Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10(Feb):441–474, 2009.

[24] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

[25] Omar Rivasplata, Emilio Parrado-Hernández, John S Shawe-Taylor, Shiliang Sun, and Csaba Szepesvári. Pac-bayes bounds for stable algorithms with instance-dependent priors. In *Advances in Neural Information Processing Systems*, pages 9214–9224, 2018.

[26] Saurabh Verma and Zhi-Li Zhang. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1539–1548, 2019.

[27] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[28] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

[29] David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 429–439. JMLR. org, 2017.

[30] Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017.

[31] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 136–145. JMLR. org, 2017.

[32] Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. *arXiv preprint arXiv:1912.02175*, 2019.

[33] Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. Deep graph matching via blackbox differentiation of combinatorial solvers. *arXiv preprint arXiv:2003.11657*, 2020.

[34] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. *arXiv preprint arXiv:2002.08676*, 2020.

[35] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[36] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[37] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[38] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

[39] Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

[40] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[41] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[42] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, pages 688–699, 2019.

[43] Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, and Armin Askari. Implicit deep learning. *arXiv preprint arXiv:1908.06315*, 2019.

A. Approximation Ability

How will the algorithm properties affect the approximation ability of deep architecture with reasoning layers? Given a model space $\mathcal{F} := \{\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) : \phi \in \Phi, \theta \in \Theta\}$, we are interested in its approximation ability to functions of the form $\texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})$. More specifically, we define the loss $\ell_{\phi,\theta}(\boldsymbol{x}, \boldsymbol{b}) := \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})\|_2$, and measure the approximation ability by $\inf_{\phi\in\Phi,\theta\in\Theta} \sup_{Q^*\in\mathcal{Q}^*} P\ell_{\phi,\theta}$, where $\mathcal{Q}^* := \{\mathcal{X} \times \mathcal{B} \mapsto \mathcal{S}_{\mu,L}^{d\times d}\}$ and $P\ell_{\phi,\theta} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{b}}[\ell_{\phi,\theta}(\boldsymbol{x}, \boldsymbol{b})]$. Intuitively, using a faster converging algorithm, the model $\texttt{Alg}_\phi^k$ could represent the reasoning-task structure, $\texttt{Opt}$, better and improve the overall approximation ability. Indeed we can prove the following lemma confirming this intuition.

Lemma A.1. (Faster Convergence ⇒ Better Approximation Ability). *Assume the problem setting in Sec 2. The approximation ability can be bounded by two terms:*

$$\inf_{\phi,\theta} \sup_{Q^*\in\mathcal{Q}^*} P\ell_{\phi,\theta} \leq \sigma_b \mu^{-2} \underbrace{\inf_\theta \sup_{Q^*} P\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F}_{\textit{approximation ability of the neural module}} + M \underbrace{\inf_{\phi\in\Phi} \mathit{Cvg}(k,\phi)}_{\textit{best convergence}}.$$

With Lemma A.1, we conclude that: A faster converging algorithm can define a model with better approximation ability. For example, for a fixed k and Q_θ, NAG converges faster than GD, so $\texttt{NAG}_\phi^k$ can approximate $\texttt{Opt}$ more accurately than $\texttt{GD}_\phi^k$, which is experimentally validated in Sec C.

Similarly, we can also reverse the reasoning, and ask the question that, given two hydrid architectures with the same approximation error, which architecture has a smaller error in representing the energy function Q^*? We show that this error is also intimately related to the convergence of the algorithm.

Lemma A.2. (Faster Convergence ⇒ Better Representation of Q^*). *Assume the problem setting in Sec 2.* $\forall \phi \in \Phi, \theta \in \Theta, Q^* \in \mathcal{Q}^* := \{\mathcal{X} \times \mathcal{B} \mapsto \mathcal{S}_{\mu,L}^{d\times d}\}$, *it holds true that*

$$P\ell_{\phi,\theta}^2 = \varepsilon \implies P\|Q_\theta - Q^*\|_F^2 \leq \sigma_b^{-2} L^4 (\sqrt{\varepsilon} + M \cdot \mathit{Cvg}(k,\phi))^2. \tag{4}$$

Lemma A.2 implies the benefit of using an algorithmic layer that aligns with the reasoning-task structure. Here the task structure is represented by $\texttt{Opt}$, the minimizer, and convergence measures how well $\texttt{Alg}_\phi^k$ is aligned with $\texttt{Opt}$. Lemma A.2 essentially indicates that *if the structure of a reasoning module can better align with the task structure, then it can better constrain the search space of the underlying neural module* Q_θ, making it easier to learn, and further lead to better sample complexity, which we will explain more in the next section.

As a concrete example for Lemma A.2, if $\texttt{GD}_\phi^k(Q_\theta, \cdot)$ and $\texttt{NAG}_\phi^k(Q_\theta, \cdot)$ achieve the **same** accuracy for approximating $\texttt{Opt}(Q^*, \cdot)$, then the neural module Q_θ in $\texttt{NAG}_\phi^k(Q_\theta, \cdot)$ will have a **better** accuracy for approximating Q^* than the Q_θ in $\texttt{GD}_\phi^k(Q_\theta, \cdot)$. In other words, a faster converging algorithm imposes more constraints on the energy function Q_θ, making it approach Q^* faster.

B. Pros and Cons for RNN as a Reasoning Layer

It has been shown that RNN (or GNN) can represent reasoning and iterative algorithms over structures [18, 14]. For example, it is proposed to use RNN to learn an optimization algorithm [18] where the update steps in each iteration are given by the operations in an RNN cell

$$\boldsymbol{y}_{k+1} \leftarrow \texttt{RNNcell}(Q, \boldsymbol{b}, \boldsymbol{y}_k) := V\sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_t + W_2^1\boldsymbol{g}_t\right)\right)\right). \tag{5}$$

In the above equation, we take a specific example where the $\texttt{RNNcell}$ is a multi-layer perception (MLP) with activations $\sigma = \text{RELU}$ that takes $\boldsymbol{y}_k$ and the gradient $\boldsymbol{g}_t = Q\boldsymbol{y}_t + \boldsymbol{b}$ as inputs. Suppose we denote $\texttt{RNN}_\phi^k$ as a recurrent neural network that has k unrolled RNN cells and view it as a neural algorithm. Can our analysis framework also be used to understand $\texttt{RNN}_\phi^k$ and how will its behavior compare with other more interpretable algorithm layers such as $\texttt{GD}_\phi^k$ and $\texttt{NAG}_\phi^k$?

We view $\texttt{RNN}_\phi^k$ as an algorithm and summarize its algorithm properties in Table 1. Assume $\phi = \{V, W_1^1, W_2^1, W^{2:L}\}$ is in a stable region $c_\phi := \sup_Q \|V\|_2 \|W_1^1 + W_2^1 Q\|_2 \prod_{l=2}^L \|W^l\|_2 < 1$, so that the operations in $\texttt{RNNcell}$ are strictly contractive, i.e., $\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|_2 < \|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|_2$. In this case, the stability and sensitivity of $\texttt{RNN}_\phi^k$ is guaranteed to be bounded. Table 1 only shows the best-case convergence, due to a fundamental disadvantage of RNN compared to GD and

NAG. For an arbitrarily fixed ϕ in the stable region, the outputs of $\texttt{RNN}_\phi^k$ with different k can form a convergent sequence, which could coincide with the outputs of $\texttt{GD}_\phi^k$ or $\texttt{NAG}_\phi^k$ with suitable choices of ϕ. However, in general the outputs of $\texttt{RNN}_\phi^k$ may not converge to the minimizer $\texttt{Opt}$. In contrast, $\texttt{GD}_\phi^k$ and $\texttt{NAG}_\phi^k$ has the worst-case convergence guarantee. This property also allows their sensitivities to decrease to 0 as k grows. In generalization analysis, the worst-case matters more, so GD and NAG are advantageous.

The advantage of RNN is its expressiveness, especially given the universal approximation ability of MLP in the RNNcell. Using existing algorithm as a reasoning layer restricts the deep model to perform a specific type of reasoning. When the needed type of reasoning is unknown or beyond what existing algorithm is capable of, RNN has the potential to learn new reasoning given sufficient data.

Table 1. Properties of $\texttt{RNN}_\phi^k$. (Details are given in Appendix G.)

Stable region Φ	$c_\phi < 1$
$\mathit{Stab}(k,\phi)$	$\mathcal{O}(1-c_\phi^k)$
$\mathit{Sens}(k)$	$\mathcal{O}(1-(\inf_\phi c_\phi)^k)$
$\min_\phi \mathit{Cvg}(k,\phi)$	$\mathcal{O}(\rho^k)$ with $\rho < 1$

C. Experimental Validation

Our experiments aim to validate our theoretical prediction with computational simulations, rather than obtaining state-of-the-art results. We hope the theory together with these experiments can lead to practical guidelines for designing deep architectures with reasoning layers.

The experiments follow the problem setting in Sec 2. 10000 pairs of $(\boldsymbol{x}, \boldsymbol{b})$ are uniformly sampled and used as the overall dataset. During training, n samples are randomly drawn from these 10000 data points as the training set. Each $Q^*(\boldsymbol{x})$ is produced by a rotation matrix and a vector of eigenvalues parameterized by a randomly fixed 2-layer dense neural network with hidden dimension 3. Then the labels are generated according to $\boldsymbol{y} = \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})$. We train the model $\texttt{Alg}_\phi^k(Q_\theta, \cdot)$ on S_n using the loss in Eq. **??**. Q_θ has the same overall architecture as Q^* but the hidden dimension could vary. Note that in all figures, each k corresponds to an **independently trained model** with k iterations in the algorithm layer, instead of the sequential outputs of a single model. Each model is trained by ADAM and SGD with learning rate grid-searched from [1e-2,5e-3,1e-3,5e-4,1e-4], and only the best result is reported. Furthermore, error bars are produced by 20 independent instantiations of the experiments. See Appendix H for more details.

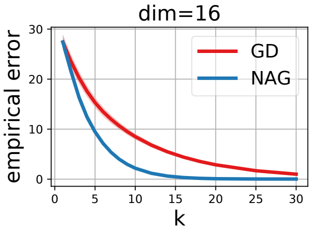

Figure 3. Approximation error.

Approximation ability. To validate Lemma A.1, we compare $\texttt{GD}_\phi^k(Q_\theta, \cdot)$ and $\texttt{NAG}_\phi^k(Q_\theta, \cdot)$ in terms of approximation accuracy. For various hidden sizes of Q_θ, the results are similar, so we report one representative in Fig 3. The approximation accuracy aligns with the convergence of the algorithms, showing that faster converging algorithm can induce better approximation ability.

Faster convergence⇒better Q_θ. We report the error of the neural module Q_θ in Fig 4. Note that $\texttt{Alg}_\phi^k(Q_\theta, \cdot)$ is trained end-to-end, without supervision on Q_θ. In Fig 4, the error of Q_θ decreases as k grows, in a rate similar to algorithm convergence. This validates the implication of Lemma A.2 that, when $\texttt{Alg}_\phi^k$ is closer to $\texttt{Opt}$, it can help the underlying neural module Q_θ to get closer to Q^*.

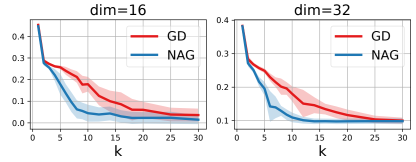

Figure 4. $P\|Q_\theta - Q^*\|_F^2$

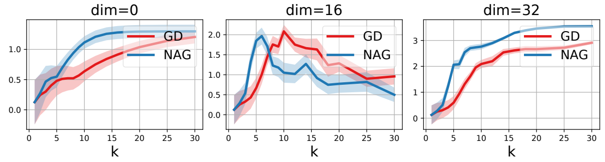

Figure 5. Generalization gap

Generalization gap. In Fig 5, we report the generalization gaps, with hidden sizes of Q_θ being 0, 16, and 32, which corresponds to the three cases **(ii)**, **(iii)**, and **(i)** discussed under Theorem 3.1, respectively. Comparing Fig 5 to Fig 2, we can see that the experimental results match very well with the theoretical implications.

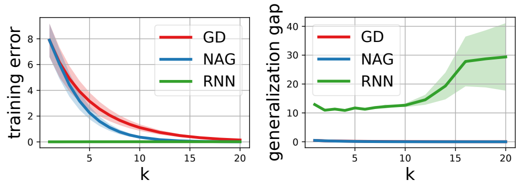

Figure 6. Algorithm layers vs RNN.

RNN. As discussed in Sec B, RNN can be viewed as neural algorithms. To have a cleaner comparison, we report their behaviors under the 'learning to optimize' senario where the objectives $(Q, \boldsymbol{b})$ are given. Fig 6 shows that RNN has a better representation power but worse generalization ability.

D. Proof of Algorithm Properties

In this section, we study several important properties of gradient descent algorithm (GD) and Nesterov's accelerated gradient algorithm (NAG), which have been summarized in Table 2. To simplify the presentation, we shall focus on quadratic minimization problems as in Section 2 and estimate the sharp dependence on the iteration number k.

Table 2. algorithm properties comparison between GD and NAG. For simplicity, only the order in k is presented. Complete statements with detailed coefficients and proofs are given in Appendix D.

`Alg`	*Cvg*(k,ϕ)	*Stab*(k,ϕ)	*Sens*(k)	Stable region Φ
$\mathtt{GD}_\phi^k$	$\mathcal{O}\left((1-\phi\mu)^k\right)$	$\mathcal{O}\left(1-(1-\phi\mu)^k\right)$	$\mathcal{O}\left(k(1-c_0\mu)^{k-1}\right)$	$[c_0, \frac{2}{\mu+L}]$
$\mathtt{NAG}_\phi^k$	$\mathcal{O}\left(k(1-\sqrt{\phi\mu})^k\right)$	$\mathcal{O}\left(1-(1-\sqrt{\phi\mu})^k\right)$	$\mathcal{O}\left(k^3(1-\sqrt{c_0\mu})^k\right)$	$[c_0, \frac{4}{\mu+3L}]$

More precisely, in the subsequent analysis, we shall fix the constants $L \geq \mu > 0$ and assume the objective function is in the function class $\mathcal{Q}_{\mu,L}$, which contains all μ-strongly convex and L-smooth quadratic functions on $\mathbb{R}^d$. Then, for any given $f \in \mathcal{Q}_{\mu,L}$, the eigenvalue decomposition enables us to represent the Hessian matrix of f, denoted by Q, as $Q = U\Lambda U^\top$, where Λ is a diagonal matrix comprising of the eigenvalues $(\lambda_i)_{i=1}^d$ of Q sorted in ascending order, i.e., $\mu \leq \lambda_1 \leq \ldots \leq \lambda_d \leq L$, and $U \in \mathbb{R}^{d\times d}$ is an orthogonal matrix whose columns constitute an orthonormal basis of corresponding eigenvectors of Q. Moreover, we shall denote by $\mathbb{I}_d$ the $d \times d$ identity matrix, and by $||A||_2$ the spectral norm of a given matrix $A \in \mathbb{R}^{d\times d}$.

We start with the GD algorithm. Let $f \in \mathcal{Q}_{\mu,L}$, $s \geq 0$ be the stepsize, and $x_0 \in \mathbb{R}^d$ be the initial guess. For each $k \in \mathbb{N} \cup \{0\}$, we denote by x_{k+1} the $k+1$-th iterate generated by the following recursive formula (cf. the output $\boldsymbol{y}_{k+1}$ of $\mathtt{GD}_\phi$ in Section 3):

$$x_{k+1} = x_k - s\nabla f(x_k). \tag{6}$$

The following theorem establishes the convergence of Eq. 6 as k tends to infinity, and the Lipschitz dependence of the iterates $(x_k^s)_{k\in\mathbb{N}}$ in terms of the stepsize s (i.e., the sensitivity of GD). Similar results can be established for general μ-strongly convex and L-smooth objective functions.

Theorem D.1. *Let $f \in \mathcal{Q}_{\mu,L}$ admit the minimiser $x^* \in \mathbb{R}^d$, $x_0 \in \mathbb{R}^d$ and for each $s \geq 0$ let $(x_k^s)_{k\in\mathbb{N}\cup\{0\}}$ be the iterates generated by Eq. 6 with stepsize s. Then we have for all $k \in \mathbb{N}$, $c_0 > 0$, $s, t \in [c_0, \frac{2}{\mu+L}]$ that*

$$\|x_k^s - x^*\|_2 \leq (1-s\mu)^k \|x_0 - x^*\|_2, \ \|x_k^t - x_k^s\|_2 \leq Lk(1-c_0\mu)^{k-1}|t-s|\|x_0 - x^*\|_2. \tag{7}$$

Proof. Let Q be the Hessian matrix of f and $(\lambda_i)_{i=1}^d$ be the eigenvalues of Q. By using the fact that $\nabla f(x^*) = 0$ and Eq. 6, we can obtain for all $k \in \mathbb{N} \cup \{0\}$ and $s \geq 0$ that $x_k^s - x^* = (\mathbb{I}_d - sQ)(x_{k-1}^s - x^*) = (\mathbb{I}_d - sQ)^k(x_0 - x^*)$.

Since the spectral norm of a matrix is invariant under orthogonal transformations, we have for all $s \in [c_0, \frac{2}{\mu+L}]$ that

$$\begin{aligned} \|\mathbb{I}_d - sQ\|_2 = \|\mathbb{I}_d - s\Lambda\|_2 &= \max_{i=1,\ldots,d} |1 - s\lambda_i| = \max(|1-s\mu|, |1-sL|) \\ &\leq 1 - s\mu. \end{aligned} \tag{8}$$

Hence, for any given $k \in \mathbb{N} \cup \{0\}$, the inequality that $\|x_k^s - x^*\|_2 \leq (\|\mathbb{I}_d - sQ\|_2)^k \|x_0 - x^*\|_2$ leads us to the desired estimate for $(\|x_k^s - x^*\|_2)_{k\in\mathbb{N}\cup\{0\}}$.

Now let $t, s \in [c_0, \frac{2}{\mu+L}]$ be given, by using the fact that $\frac{d}{ds}x_k^s = k(\mathbb{I}_d - sQ)^{k-1}Q(x_0 - x^*)$ for all $s > 0$, we can deduce from the mean value theorem that

$$\begin{aligned} \|x_k^s - x_k^t\|_2 &\leq \left(\sup_{r\in(c_0, \frac{2}{\mu+L})} \|\tfrac{d}{dr}x_k^r\|_2 \right) |t-s| \\ &\leq \left(\sup_{r\in(c_0, \frac{2}{\mu+L})} k(\|\mathbb{I}_d - rQ\|_2)^{k-1} \|Q\|_2 \|x_0 - x^*\|_2 \right) |t-s| \\ &\leq k \left(\sup_{r\in[c_0, \frac{2}{\mu+L}]} \|\mathbb{I}_d - rQ\|_2 \right)^{k-1} L|t-s|\|x_0 - x^*\|_2, \end{aligned}$$

which along with Eq. 8 finishes the proof of the desired sensitivity estimate. □

The next theorem shows that Eq. 6 with stepsize $s \in (0, \frac{2}{\mu+L}]$ is Lipschitz stable in terms of the perturbations of f. In particular, for a quadratic function $f \in \mathcal{Q}_{\mu,L}$, we shall establish the Lipschitz stability with respect to the perturbations in the parameters of f. For notational simplicity, we assume $x_0 = 0$ as in Section 3, but it is straightforward to extend the results to an arbitrary initial guess $x_0 \in \mathbb{R}^d$.

Theorem D.2. *Let $x_0 = 0$, for each $i \in \{1, 2\}$ let $f_i \in \mathcal{Q}_{\mu,L}$ admit the minimizer $x^{*,i} \in \mathbb{R}^d$ and satisfy $\nabla f_i(x) = Q_i x + b_i$ for a symmetric matrix $Q_i \in \mathbb{R}^{d\times d}$ and $b_i \in \mathbb{R}^d$, for each $i \in \{1, 2\}$, $s > 0$ let $(x^s_{k,i})_{k\in\mathbb{N}\cup\{0\}}$ be the iterates generated by Eq. 6 with $f = f_i$ and stepsize s, and let $M = \min(\|x^{*,1}\|_2, \|x^{*,2}\|_2)$. Then we have for all $k \in \mathbb{N}$, $c_0 > 0$, $s \in [c_0, \frac{2}{\mu+L}]$ that:*

$$
\begin{aligned}
\|x^s_{k,1} - x^s_{k,2}\|_2 \le & \left[\frac{1}{\mu}\big(1-(1-s\mu)^k\big) + sk(1-s\mu)^{k-1}\right] M\|Q_1 - Q_2\|_2 \\
& + \frac{1}{\mu}\big(1-(1-s\mu)^k\big)\|b_1 - b_2\|_2.
\end{aligned}
$$

Proof. Let us assume without loss of generality that $\|x^{*,2}\|_2 \le \|x^{*,1}\|_2$ and $c_0 \le \frac{2}{\mu+L}$. We write $\delta x_k = x^s_{k,1} - x^s_{k,2}$ for each $k \in \mathbb{N}\cup\{0\}$. Then, by using Eq. 6 and the fact that $\nabla f_1(x) - \nabla f_1(y) = Q_1(x-y)$ for all $x, y \in \mathbb{R}^d$, we can deduce that $\delta x_0 = 0$ and for all $k \in \mathbb{N}\cup\{0\}$ that

$$
\delta x_{k+1} = (\mathbb{I}_d - sQ_1)\delta x_k + e_k = \sum_{i=0}^{k} (\mathbb{I}_d - sQ_1)^i e_{k-i},
$$

where $e_k = -s(\nabla f_1 - \nabla f_2)(x^s_{k,2})$ for each $k \in \mathbb{N}\cup\{0\}$. Note that it holds for all $k \in \mathbb{N}\cup\{0\}$ that

$$
\begin{aligned}
\|e_k\|_2 \le s\|(\nabla f_1 - \nabla f_2)(x^s_{k,2})\|_2 &\le s\left(\|Q_2 - Q_2\|_2\|x^s_{k,2}\|_2 + \|b_1 - b_2\|_2\right) \\
&\le s\big(\|Q_2 - Q_2\|_2(\|x^{*,2}\|_2 + \|x^s_{k,2} - x^{*,2}\|_2) + \|b_1 - b_2\|_2\big) \\
&\le s\big(\|Q_2 - Q_2\|_2(\|x^{*,2}\|_2 + (1-s\mu)^k\|x_0 - x^{*,2}\|_2) + \|b_1 - b_2\|_2\big),
\end{aligned}
$$

where we have applied Theorem D.1 for the last inequality. Thus for each $k \in \mathbb{N}$, we can obtain from Eq. 8 and $x_0 = 0$ that

$$
\begin{aligned}
\|\delta x_k\|_2 &\le \sum_{i=0}^{k-1} (\|\mathbb{I}_d - sQ_1\|_2)^i \|e_{k-1-i}\|_2 \\
&\le \sum_{i=0}^{k-1} (1-s\mu)^i s\big[(1 + (1-s\mu)^{k-1-i})\|x^{*,2}\|_2\|Q_2 - Q_2\|_2 + \|b_1 - b_2\|_2\big] \\
&= \left[\frac{1}{\mu}\big(1-(1-s\mu)^k\big) + sk(1-s\mu)^{k-1}\right] \min(\|x^{*,1}\|_2, \|x^{*,2}\|_2)\|Q_2 - Q_2\|_2 \\
&\quad + \frac{1}{\mu}\big(1-(1-s\mu)^k\big)\|b_1 - b_2\|_2.
\end{aligned}
$$

which leads to the desired conclusion due to the fact that $M = \min(\|x^{*,1}\|_2, \|x^{*,2}\|_2)$. □

We now proceed to investigate similar properties of the NAG algorithm, whose proofs are more involved due to the fact that NAG is a multi-step method.

Recall that for any given $f \in \mathcal{Q}_{\mu,L}$, initial guess $x_0 \in \mathbb{R}^d$ and stepsize $s \ge 0$, the NAG algorithm generates iterates $(x_k, y_k)_{k\in\mathbb{N}\cup\{0\}}$ as follows: $y_0 = x_0$ and for each $k \in \mathbb{N}\cup\{0\}$,

$$
x_{k+1} = y_k - s\nabla f(y_k), \quad y_{k+1} = x_{k+1} + \frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}}(x_{k+1} - x_k). \tag{9}
$$

Note that x_{k+1}, y_{k+1} are denoted by $\boldsymbol{y}_{k+1}, \boldsymbol{z}_{k+1}$, respectively, in Section 3.

We first introduce the following matrix $R_{\text{NAG},s}$ for Eq. 9 for any given function $f \in \mathcal{Q}_{\mu,L}$ and stepsize $s \in [0, \frac{4}{3L+\mu}]$:

$$R_{\text{NAG},s} \coloneqq \begin{pmatrix} (1+\beta_s)(\mathbb{I}_d - sQ) & -\beta_s(\mathbb{I}_d - sQ) \\ \mathbb{I}_d & 0 \end{pmatrix} \tag{10}$$

where $\beta_s = \frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}}$ and Q is the Hessian matrix of f. The following lemma establishes an upper bound of the spectral norm of the k-th power of $R_{\text{NAG},s}$, which extends [17, Lemma 22] to block matrices, a wider range of stepsize (s is allowed to be larger than $1/L$) and a momentum parameter β_s depending on the stepsize s.

Lemma D.1. *Let $f \in \mathcal{Q}_{\mu,L}$, $s \in (0, \frac{4}{3L+\mu}]$, $\beta_s = \frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}}$ and $R_{NAG,s}$ be defined as in Eq. 10. Then we have for all $k \in \mathbb{N}$ that $\|R^k_{NAG,s}\|_2 \leq 2(k+1)(1-\sqrt{\mu s})^k$.*

Proof. Let $Q = U\Lambda U^T$ be the eigenvalue decomposition of the Hessian matrix Q of f, where Λ is a diagonal matrix comprising of the corresponding eigenvalues of Q sorted in ascending order, i.e., $0 < \mu \leq \lambda_1 \leq \ldots \leq \lambda_d \leq L$. Then we have that

$$R_{\text{NAG},s} = \begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix} \begin{pmatrix} (1+\beta_s)(\mathbb{I}_d - s\Lambda) & -\beta_s(\mathbb{I}_d - s\Lambda) \\ \mathbb{I}_d & 0 \end{pmatrix} \begin{pmatrix} U^T & 0 \\ 0 & U^T \end{pmatrix},$$

which together with the facts that any permutation matrix is orthogonal, and the spectral norm of a matrix is invariant under orthogonal transformations, gives us the identity that: for all $k \in \mathbb{N}$,

$$\|R^k_{\text{NAG},s}\|_2 = \left|\left|\left| \begin{pmatrix} (1+\beta_s)(\mathbb{I}_d - s\Lambda) & -\beta_s(\mathbb{I}_d - s\Lambda) \\ \mathbb{I}_d & 0 \end{pmatrix}^k \right|\right|\right|_2 = \max_{i=1,\ldots n} \|T^k_{s,i}\|_2, \tag{11}$$

where $T_{s,i} = \left(\begin{smallmatrix} (1+\beta_s)(1-s\lambda_i) & -\beta_s(1-s\lambda_i) \\ 1 & 0 \end{smallmatrix}\right)$ for all $i = 1, \ldots, d$.

Now let $s \in (0, \frac{4}{3L+\mu}]$ and $i = 1, \ldots, d$ be fixed. If $1 - s\lambda_i \geq 0$, by using [**?**]Lemma 22]chen2018stability (with $\alpha = \mu$, $\beta = 1/s$, $h = 1 - s\lambda_i$ and $\kappa = \beta/\alpha = 1/(\mu s)$), we can obtain that

$$\|T^k_{s,i}\|_2 \leq 2(k+1)\left(\frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}}(1-\mu s)\right)^{k/2} \leq 2(k+1)(1-\sqrt{\mu s})^k.$$

We then discuss the case where $1 - s\lambda_i < 0$. Let us write $T^k_{s,i} = \left(\begin{smallmatrix} a_k & b_k \\ c_k & d_k \end{smallmatrix}\right)$ for each $k \in \mathbb{N} \cup \{0\}$, then we have for all $k \in \mathbb{N}$ that

$$a_k = (1+\beta_s)(1-s\lambda_i)a_{k-1} - \beta_s(1-s\lambda_i)c_{k-1}, \quad c_k = a_{k-1},$$
$$b_k = (1+\beta_s)(1-s\lambda_i)b_{k-1} - \beta_s(1-s\lambda_i)d_{k-1}, \quad d_k = b_{k-1},$$

with $a_1 = (1+\beta_s)(1-s\lambda_i)$, $b_1 = -\beta_s(1-s\lambda_i)$, $c_1 = 1$ and $d_1 = 0$. Since the conditions $1 - s\lambda_i < 0$ and $s \leq \frac{4}{3L+\mu}$ imply that $\lambda_i > \frac{1}{s} \geq \frac{3L+\mu}{4} \geq \mu$, we see the discriminant of the characteristic polynomial satisfies that

$$\Delta = (1+\beta_s)^2(1-s\lambda_i)^2 - 4\beta_s(1-s\lambda_i) = \frac{4(1-s\lambda_i)}{(1+\sqrt{\mu s})^2}s(\mu - \lambda_i) > 0,$$

which implies that there exist $l_1, l_2, l_3, l_4 \in \mathbb{R}$ such that it holds for all $k \in \mathbb{N} \cup \{0\}$ that $a_k = l_1\tau_+^{k+1} + l_2\tau_-^{k+1}$ and $b_k = l_3\tau_+^{k+1} + l_4\tau_-^{k+1}$, with $\tau_\pm = \frac{(1+\beta_s)(1-s\lambda_i)\pm\sqrt{\Delta}}{2}$, $l_1 = \frac{1}{\tau_+-\tau_-}$, $l_2 = -\frac{1}{\tau_+-\tau_-}$, $l_3 = \frac{-\tau_-}{\tau_+-\tau_-}$ and $l_4 = \frac{\tau_+}{\tau_+-\tau_-}$. Thus, by letting $\rho_i \coloneqq \max(|\tau_+|, |\tau_-|)$, we have that $|a_k| = |\sum_{j=0}^k \tau_+^{k-j}\tau_-^j| \leq (k+1)\rho_i^k$ and $|b_k| = |(-\tau_+\tau_-)\sum_{j=0}^{k-1}\tau_+^{k-1-j}\tau_-^j| \leq k\rho_i^{k+1}$ for all $k \in \mathbb{N} \cup \{0\}$.

Now we claim that the conditions $1 - s\lambda_i < 0$ and $0 < s \leq \frac{4}{3L+\mu}$ imply the estimate that $\rho_i \leq 1 - \sqrt{\mu s} < 1$. In fact, the inequality $s \leq \frac{4}{3L+\mu}$ gives us that $\mu s \leq \frac{4\mu}{3L+\mu} \leq 1$, which implies that $\beta_s = \frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}} \geq 0$. Hence we can deduce from $1 - s\lambda_i < 0$ that $\sqrt{\Delta} \geq (1+\beta_s)(s\lambda_i - 1)$ and

$$|\tau_+| \leq |\tau_-| \leq \frac{s\lambda_i - 1 + \sqrt{(s\lambda_i - 1)s(\lambda_i - \mu)}}{1+\sqrt{\mu s}} \leq \frac{sL - 1 + \sqrt{(sL-1)s(L-\mu)}}{1+\sqrt{\mu s}}.$$

Note that $2-(\mu+L)s \geq 2-\frac{4(\mu+L)}{3L+\mu} \geq 0$, we see that

$$\begin{aligned}\rho_i \leq 1-\sqrt{\mu s} \impliedby |\tau_-| \leq 1-\sqrt{\mu s} &\impliedby sL-1+\sqrt{(sL-1)s(L-\mu)} \leq 1-\mu s \\ &\iff (sL-1)s(L-\mu) \leq (2-(\mu+L)s)^2 \\ &\iff (us-1)((3L+\mu)s-4) \geq 0.\end{aligned}$$

Therefore, we have that $\max(|a_k|,|b_k|,|c_k|,|d_k|) \leq (k+1)(1-\sqrt{\mu s})^k$, which, along with the relationship between the spectral norm and Frobenius norm, gives us that $\|T^k_{s,i}\|_2 \leq \|T^k_{s,i}\|_{\mathrm{F}} \leq 2(k+1)(1-\sqrt{\mu s})^k$, and finishes the proof of the desired estimate for the case with $1-s\lambda_i<0$. □

As an important consequence of Lemma D.1, we now obtain the following upper bound of the error $(\|x_k-x^*\|_2)_{k\in\mathbb{N}}$ for any given objective function $f\in\mathcal{Q}_{\mu,L}$ and stepsize $s\in(0,\frac{4}{3L+\mu}]$.

Theorem D.3. *Let $f\in\mathcal{Q}_{\mu,L}$ admit the minimizer $x^*\in\mathbb{R}^d$, $x_0\in\mathbb{R}^d$, $s\in(0,\frac{4}{3L+\mu}]$ and $(x^s_k,y^s_k)_{k\in\mathbb{N}\cup\{0\}}$ be the iterates generated by Eq. 9 with stepsize s. Then we have for all $k\in\mathbb{N}\cup\{0\}$ that*

$$\|x^s_{k+1}-x^*\|_2^2+\|x^s_k-x^*\|_2^2 \leq 8(1+k)^2(1-\sqrt{\mu s})^{2k}\|x_0-x^*\|_2^2.$$

Proof. For any $f\in\mathcal{Q}_{\mu,L}$, and $s\in(0,\frac{4}{3L+\mu}]$, by letting $\beta_s=\frac{1-\sqrt{\mu s}}{1+\sqrt{\mu s}}$, we can rewrite Eq. 9 as follows: $x^s_0=x_0$, $x^s_1=x_0-s\nabla f(x_0)$ and for all $k\in\mathbb{N}$,

$$x^s_{k+1}=(1+\beta_s)x^s_k-\beta_s x_{k-1}-s\nabla f((1+\beta_s)x^s_k-\beta_s x_{k-1}), \tag{12}$$

which together with the fact that $\nabla f(x^*)=0$ shows that

$$\begin{pmatrix} x^s_{k+1}-x^* \\ x^s_k-x^*\end{pmatrix} = R_{\text{NAG},s}\begin{pmatrix} x^s_k-x^* \\ x^s_{k-1}-x^*\end{pmatrix} = R^k_{\text{NAG},s}\begin{pmatrix} x^s_1-x^* \\ x^s_0-x^*\end{pmatrix}$$

where $R_{\text{NAG},s}$ is defined as in Eq. 10. Hence by using $x^s_1=x_0-s\nabla f(x_0)$ and Theorem D.1, we can obtain that

$$\begin{aligned}\|x^s_{k+1}-x^*\|_2^2+\|x^s_k-x^*\|_2^2 &\leq \|R^k_{\text{NAG},s}\|_2^2(\|x^s_1-x^*\|_2^2+\|x^s_0-x^*\|_2^2) \\ &\leq \|R^k_{\text{NAG},s}\|_2^2 2\|x_0-x^*\|_2^2,\end{aligned}$$

which together with Lemma D.1 leads to the desired convergence result. □

Remark D.1. It is well-known that for a general μ-strongly convex and L-smooth objective function f, one can employ a Lyapunov argument and establish that the iterates obtained by Eq. 9 with stepsize $s\in[0,\frac{1}{L}]$ satisfy the estimate that $\|x_k-x^*\|_2^2 \leq \frac{2L}{\mu}(1-\sqrt{\mu s})^k\|x_0-x^*\|_2^2$. Here by taking advantage of the affine structure of ∇f, we have obtained a sharper estimate of the convergence rate for a wider range of stepsize $s\in(0,\frac{4}{3L+\mu}]$.

We also would like to emphasize that the upper bound in Theorem D.3 is tight, in the sense that the additional quadratic dependence on k in the error estimate is inevitable. In fact, one can derive a *closed-form expression* of $R^k_{\text{NAG},s}$ and show that, for an index i such that the eigenvalue λ_i is sufficiently close to μ, the squared error for that component is of the magnitude $\mathcal{O}((k\sqrt{\mu s}+1)^2(1-\sqrt{\mu s})^{2k})$.

We then proceed to analyze the sensitivity of Eq. 9 with respect to the stepsize. The following theorem shows that the iterates $(x_k,y_k)_{k\in\mathbb{N}\cup\{0\}}$ generated by Eq. 9 depend Lipschitz continuously on the stepsize s.

Theorem D.4. *Let $f\in\mathcal{Q}_{\mu,L}$ admit the minimiser $x^*\in\mathbb{R}^d$, $x_0\in\mathbb{R}^d$, and for each $s\in(0,\frac{4}{3L+\mu}]$ let $(x^s_k,y^s_k)_{k\in\mathbb{N}\cup\{0\}}$ be the iterates generated by Eq. 9 with stepsize s. Then we have for all $k\in\mathbb{N}$, $c_0>0$ and $t,s\in[c_0,\frac{4}{3L+\mu}]$ that:*

$$\|x^t_k-x^s_k\|_2 \leq \left(2L(1+k)+\frac{4}{3}k(k+1)(k+5)\left(\sqrt{\frac{\mu}{c_0}}+2L\right)\right)(1-\sqrt{\mu c_0})^k|t-s|\|x_0-x^*\|_2.$$

Proof. Throughout this proof we assume without loss of generality that $c_0 \le s < t \le \frac{4}{3L+\mu}$. Let Q be the Hessian matrix of f, for each $r \in [c_0, \frac{4}{3L+\mu}]$ let $\beta_r = \frac{1-\sqrt{\mu r}}{1+\sqrt{\mu r}}$, and for each $k \in \mathbb{N} \cup \{0\}$ let $\delta x_k = x_k^t - x_k^s$. Then we can deduce from Eq. 12 that $\delta x_0 = 0$, $\delta x_1 = -(t-s)\nabla f(x_0)$ and for all $k \in \mathbb{N}$ that

$$\begin{aligned} x_{k+1}^t - x_{k+1}^s = {} & [(1+\beta_t)x_k^t - \beta_t x_{k-1}^t - t\nabla f((1+\beta_t)x_k^t - \beta_t x_{k-1}^t)] \\ & - [(1+\beta_s)x_k^s - \beta_s x_{k-1}^s - s\nabla f((1+\beta_s)x_k^s - \beta_s x_{k-1}^s)], \end{aligned}$$

which together with the fact that $\nabla f(x) - \nabla f(y) = Q(x-y)$ for all $x, y \in \mathbb{R}^d$ shows that

$$\begin{pmatrix} \delta x_{k+1} \\ \delta x_k \end{pmatrix} = R_{\text{NAG},t} \begin{pmatrix} \delta x_k \\ \delta x_{k-1} \end{pmatrix} + \begin{pmatrix} e_k \\ 0 \end{pmatrix}$$

with $R_{\text{NAG},t}$ defined as in Eq. 10 and the following residual term

$$\begin{aligned} e_k \coloneqq {} & [(1+\beta_t)x_k^s - \beta_t x_{k-1}^s - t\nabla f((1+\beta_t)x_k^s - \beta_t x_{k-1}^s)] \\ & - [(1+\beta_s)x_k^s - \beta_s x_{k-1}^s - s\nabla f((1+\beta_s)x_k^s - \beta_s x_{k-1}^s)]. \end{aligned}$$

Hence we can obtain by induction that: for all $k \in \mathbb{N}$,

$$\begin{pmatrix} \delta x_{k+1} \\ \delta x_k \end{pmatrix} = R_{\text{NAG},t}^k \begin{pmatrix} \delta x_1 \\ \delta x_0 \end{pmatrix} + \sum_{i=0}^{k-1} R_{\text{NAG},t}^i \begin{pmatrix} e_{k-i} \\ 0 \end{pmatrix}. \tag{13}$$

Now the facts that $\nabla f(x^*) = 0$ and $\nabla^2 f \equiv Q$ gives us that

$$\begin{aligned} e_k &= (\beta_t - \beta_s)(x_k^s - x_{k-1}^s) - t\nabla f((1+\beta_t)x_k^s - \beta_t x_{k-1}^s) + s\nabla f((1+\beta_s)x_k^s - \beta_s x_{k-1}^s) \\ &= (\beta_t - \beta_s)\big((x_k^s - x^*) - (x_{k-1}^s - x^*)\big) - tQ\big((1+\beta_t)(x_k^s - x^*) - \beta_t(x_{k-1}^s - x^*)\big) \\ &\quad + sQ\big((1+\beta_s)(x_k^s - x^*) - \beta_s(x_{k-1}^s - x^*)\big) \\ &= \big[(\beta_t - \beta_s) - (t + t\beta_t - s - s\beta_s)Q\big](x_k^s - x^*) - \big[(\beta_t - \beta_s) - (t\beta_t - s\beta_s)Q\big](x_{k-1}^s - x^*). \end{aligned}$$

Note that one can easily verify that the function $g_1(r) = \beta_r$ is $\sqrt{\mu/c_0}$-Lipschitz on $[c_0, \frac{4}{3L+\mu}]$, and the function $g_2(r) = r\beta_r$ is 1-Lipschitz on $[0, \frac{4}{3L+\mu}]$. Moreover, the fact that $f \in \mathcal{Q}_{\mu,L}$ implies that $\|Q\|_2 \le L$. Thus we can obtain from Theorem D.3 that

$$\begin{aligned} \|e_k\|_2 &\le \left(\sqrt{\frac{\mu}{c_0}} + 2L\right)|t-s|\|x_k^s - x^*\|_2 + \left(\sqrt{\frac{\mu}{c_0}} + L\right)|t-s|\|x_{k-1}^s - x^*\|_2 \\ &\le \left(\sqrt{\frac{\mu}{c_0}} + 2L\right)|t-s|\sqrt{2(\|x_k^s - x^*\|_2^2 + \|x_{k-1}^s - x^*\|_2^2)} \\ &\le \left(\sqrt{\frac{\mu}{c_0}} + 2L\right)|t-s|4(1+k)(1-\sqrt{\mu s})^k\|x_0 - x^*\|_2. \end{aligned}$$

This, along with Eq. 13, Lemma D.1 and $s < t$, gives us that

$$\begin{aligned} &\sqrt{\|\delta x_{k+1}\|_2^2 + \|\delta x_k\|_2^2} \le \|R_{\text{NAG},t}^k\|_2\|\delta x_1\|_2 + \sum_{i=0}^{k-1}\|R_{\text{NAG},t}^i\|_2\|e_{k-i}\|_2 \\ &\le 2(1+k)(1-\sqrt{\mu t})^k|t-s|L\|x_0 - x^*\|_2 \\ &\quad + \sum_{i=0}^{k-1} 2(1+i)(1-\sqrt{\mu t})^i\left(\sqrt{\frac{\mu}{c_0}} + 2L\right)|t-s|4(1+k-i)(1-\sqrt{\mu s})^{k-i}\|x_0 - x^*\|_2 \\ &= \left(2L(1+k) + \frac{4}{3}k(k+1)(k+5)\left(\sqrt{\frac{\mu}{c_0}} + 2L\right)\right)|t-s|(1-\sqrt{\mu s})^k\|x_0 - x^*\|_2, \end{aligned}$$

which finishes the proof of the desired estimate due to the fact that $s \ge c_0$. □

The next theorem is an an analog of Theorem D.2 for the NAC scheme Eq. 9, which shows that the outputs of Eq. 9 with stepsize $s \in (0, \frac{4}{3L+\mu}]$ is Lipschitz stable with respect to the perturbations of the parameters in f.

Theorem D.5. *Let $x_0 = 0$, for each $i \in \{1, 2\}$ let $f_i \in \mathcal{Q}_{\mu,L}$ admit the minimizer $x^{*,i} \in \mathbb{R}^d$ and satisfy $\nabla f_i(x) = Q_i x + b_i$ for a symmetric matrix $Q_i \in \mathbb{R}^{d\times d}$ and $b_i \in \mathbb{R}^d$, for each $i \in \{1, 2\}$, $s > 0$ let $(x^s_{k,i})_{k\in\mathbb{N}\cup\{0\}}$ be the iterates generated by Eq. 9 with $f = f_i$ and stepsize s, and let $M = \min(\|x^{*,1}\|_2, \|x^{*,2}\|_2)$. Then we have for all $k \in \mathbb{N}$, $s \in [c_0, \frac{4}{3L+\mu}]$ that:*

$$
\begin{aligned}
\|x^s_{k,1} - x^s_{k,2}\|_2 \le &\left[\frac{2}{\mu}\left(1-(1-\sqrt{\mu s})^{k-1}\right) + s\frac{8(k-1)k(k+4)}{3}(1-\sqrt{\mu s})^{k-1}\right] M\|Q_1 - Q_2\|_2 \\
&+ \frac{2}{\mu}\left(1-(1-\sqrt{\mu s})^{k}\right)\|b_1 - b_2\|_2.
\end{aligned}
$$

Proof. Let us assume without loss of generality that $\|x^{*,2}\|_2 \le \|x^{*,1}\|_2$. We first fix an arbitrary $s \in [c_0, \frac{4}{3L+\mu}]$ and write $\delta x_k = x^s_{k,1} - x^s_{k,2}$ for each $k \in \mathbb{N}\cup\{0\}$. Then, by using Eq. 12 and the fact that $\nabla f_1(x) - \nabla f_1(y) = Q_1(x - y)$ for all $x, y \in \mathbb{R}^d$, we can deduce that $\delta x_0 = 0$, $\delta x_1 = -s(\nabla f_1 - \nabla f_2)(x_0)$ and for all $k \in \mathbb{N}$,

$$
\begin{pmatrix}\delta x_{k+1}\\ \delta x_k\end{pmatrix} = R_{\text{NAG},s}\begin{pmatrix}\delta x_k\\ \delta x_{k-1}\end{pmatrix} + \begin{pmatrix}e_k\\ 0\end{pmatrix} = R^k_{\text{NAG},s}\begin{pmatrix}\delta x_1\\ \delta x_0\end{pmatrix} + \sum_{j=0}^{k-1} R^j_{\text{NAG},s}\begin{pmatrix}e_{k-j}\\ 0\end{pmatrix}, \tag{14}
$$

where $R_{\text{NAG},s}$ is defined as in Eq. 10 (with $Q = Q_1$) and the residual term e_k is given by

$$
e_k := -s(\nabla f_1 - \nabla f_2)((1+\beta_s)x^s_{k,2} - \beta_s x^s_{k-1,2}) \quad \forall k \in \mathbb{N}.
$$

Note that, by using Theorem D.3 and the inequality that $x + y \le \sqrt{2(x^2 + y^2)}$ for all $x, y \in \mathbb{R}$, we have for each $k \in \mathbb{N}$ that

$$
\begin{aligned}
\|e_k\|_2 &= s\|(Q_1 - Q_2)((1+\beta_s)x^s_{k,2} - \beta_s x^s_{k-1,2}) + (b_1 - b_2)\|_2 \\
&\le s\|Q_1 - Q_2\|_2(\|x^{*,2}\|_2 + 2\|x^s_{k,2} - x^{*,2}\|_2 + \|x^s_{k-1,2} - x^{*,2}\|_2) + s\|b_1 - b_2\|_2 \\
&\le s\|Q_1 - Q_2\|_2(\|x^{*,2}\|_2 + 2\|x^s_{k,2} - x^{*,2}\|_2 + \|x^s_{k-1,2} - x^{*,2}\|_2) + s\|b_1 - b_2\|_2 \\
&\le s\|Q_1 - Q_2\|_2(\|x^{*,2}\|_2 + 8(1+k)(1-\sqrt{\mu s})^k\|x_0 - x^{*,2}\|_2) + s\|b_1 - b_2\|_2.
\end{aligned}
$$

Hence we can obtain from Eq. 14, Lemma D.1 and $x_0 = 0$ that

$$
\begin{aligned}
&\sqrt{\|\delta x_{k+1}\|_2^2 + \|\delta x_k\|_2^2} \le 2(k+1)(1-\sqrt{\mu s})^k\|\delta x_1\|_2 + \sum_{j=0}^{k-1} 2(j+1)(1-\sqrt{\mu s})^j\|e_{k-j}\|_2 \\
&\le 2(k+1)(1-\sqrt{\mu s})^k s\|b_1 - b_2\|_2 + \sum_{j=0}^{k-1} 2(j+1)(1-\sqrt{\mu s})^j\big[s\|b_1 - b_2\|_2 \\
&\quad + s\|Q_1 - Q_2\|_2(1 + 8(1+k-j)(1-\sqrt{\mu s})^{k-j})\|x^{*,2}\|_2\big] \\
&\le 2s\sum_{j=0}^{k}(j+1)(1-\sqrt{\mu s})^j\|b_1 - b_2\|_2 + 2s\sum_{j=0}^{k-1}\big[(j+1)(1-\sqrt{\mu s})^j \\
&\quad + 8(j+1)(1+k-j)(1-\sqrt{\mu s})^k\big]\|Q_1 - Q_2\|_2\min(\|x^{*,1}\|_2, \|x^{*,2}\|_2).
\end{aligned}
$$

Let $p = 1 - \sqrt{\mu s} \in [0, 1)$, then we can easily show for each $k \in \mathbb{N}\cup\{0\}$ that $(1-p)\sum_{j=0}^{k}(j+1)p^j = \sum_{j=0}^{k} p^j - p^{k+1}$, which implies that $\sum_{j=0}^{k}(j+1)(1-\sqrt{\mu s})^j \le \frac{1-(1-\sqrt{\mu s})^{k+1}}{\mu s}$. Moreover, we have that $\sum_{j=0}^{k-1}(j+1)(1+k-j) = \frac{k(k+1)(k+5)}{6}$ for all $k \in \mathbb{N}$. Thus we can simplify the above estimate and deduce for each $k \in \mathbb{N}$ that

$$
\begin{aligned}
\|\delta x_{k+1}\|_2 \le \frac{2}{\mu}\left(1-(1-\sqrt{\mu s})^{k+1}\right)\|b_1 - b_2\|_2 + &\left[\frac{2}{\mu}\left(1-(1-\sqrt{\mu s})^{k}\right)\right. \\
&\left. + s\frac{8k(k+1)(k+5)}{3}(1-\sqrt{\mu s})^k\right]\|Q_1 - Q_2\|_2\min(\|x^{*,1}\|_2, \|x^{*,2}\|_2).
\end{aligned}
$$

Moreover, the condition that $s \le \frac{4}{3L+\mu} \le \frac{1}{\mu}$ implies that $\|\delta x_1\|_2 = s\|b_1 - b_2\|_2 \le \frac{2}{\mu}\left(1-(1-\sqrt{\mu s})\right)\|b_1 - b_2\|_2$, which shows that the same upper bound also holds for $\|\delta x_1\|_2$ and finishes the proof of the desired estimate. □

E. Proof of Approximation Ability

Lemma A.1. (Faster Convergence ⇒ Better Approximation Ability). *Assume the problem setting in Sec 2. The approximation ability can be bounded by two terms:*

$$\inf_{\phi,\theta}\sup_{Q^*\in\mathcal{Q}^*} P\ell_{\phi,\theta} \leq \sigma_b\mu^{-2}\underbrace{\inf_{\theta}\sup_{Q^*} P\|Q_\theta(\boldsymbol{x})-Q^*(\boldsymbol{x})\|_F}_{\textit{approximation ability of the neural module}} + M\underbrace{\inf_{\phi\in\Phi} \mathit{Cvg}(k,\phi)}_{\textit{best convergence}}.$$

Proof. For each $\phi\in\Phi, \theta\in\Theta$, $Q^*\in\mathcal{Q}^*$,

$$\begin{aligned}
\ell_{\phi,\theta}(\boldsymbol{x},\boldsymbol{b}) &= \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}),\boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}),\boldsymbol{b})\|_2 && (15)\\
&\leq \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}),\boldsymbol{b}) - \texttt{Opt}(Q_\theta(\boldsymbol{x}),\boldsymbol{b})\|_2 + \|\texttt{Opt}(Q_\theta(\boldsymbol{x}),\boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}),\boldsymbol{b})\|_2 && (16)\\
&\leq \mathit{Cvg}(k,\phi)\|\texttt{Alg}_\phi^0(Q_\theta(\boldsymbol{x}),\boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}),\boldsymbol{b})\|_2 + \|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2 && (17)\\
&\leq \mathit{Cvg}(k,\phi)\cdot M + \|\left(Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1}\right)\boldsymbol{b}\|_2, && (18)
\end{aligned}$$

where in the last inequality we have used the facts that the initialization is assumed to be zero vector, i.e., $\texttt{Alg}_\phi^0(Q_\theta(\boldsymbol{x}),\boldsymbol{b}) = \boldsymbol{0}$, and that $M\geq \sup_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{b}\in\mathcal{B}}\texttt{Opt}(Q^*(\boldsymbol{x}),\boldsymbol{b})$. Note that the independence of $(\boldsymbol{x},\boldsymbol{b})$ and the fact that $\mathbb{E}\boldsymbol{b}\boldsymbol{b}^\top = \sigma_b^2 I$ imply that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{b}}\|\left(Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1}\right)\boldsymbol{b}\|_2^2 && (19)\\
&= \mathrm{Tr}\left((Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1})^\top(Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1})\sigma_b^2 I\right) && (20)\\
&= \sigma_b^2\|Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1}\|_F^2 && (21)\\
&= \sigma_b^2\|Q_\theta(\boldsymbol{x})^{-1}(Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x}))Q^*(\boldsymbol{x})^{-1}\|_F^2 && (22)\\
&\leq \mu^{-4}\sigma_b^2\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F^2 && (23)
\end{aligned}$$

Therefore, we see from Hölder's inequality that

$$\mathbb{E}_{\boldsymbol{b}}\|\left(Q_\theta(\boldsymbol{x})^{-1} - Q^*(\boldsymbol{x})^{-1}\right)\boldsymbol{b}\|_2 \leq \mu^{-2}\sigma_b\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F. \tag{24}$$

Collecting all the above inequalities, we have

$$P\ell_{\phi,\theta} \leq \mathit{Cvg}(k,\phi)\cdot M + \sigma_b\mu^{-2}P\|Q_\theta - Q^*\|_F. \tag{25}$$

Taking supremum over Q^*, we have

$$\sup_{Q^*\in\mathcal{Q}^*} P\ell_{\phi,\theta} \leq \mathit{Cvg}(k,\phi)\cdot M + \sigma_b\mu^{-2}\sup_{Q^*\in\mathcal{Q}^*} P\|Q_\theta - Q^*\|_F. \tag{26}$$

Taking infimum over ϕ and θ, we have

$$\inf_{\phi\in\Phi,\theta\in\Theta}\sup_{Q^*\in\mathcal{Q}^*} P\ell_{\phi,\theta} \leq \inf_{\phi\in\Phi}\mathit{Cvg}(k,\phi)\cdot M + \sigma_b\mu^{-2}\inf_{\theta\in\Theta}\sup_{Q^*\in\mathcal{Q}^*} P\|Q_\theta - Q^*\|_F. \tag{27}$$

□

Lemma A.2. (Faster Convergence ⇒ Better Representation of Q^*). *Assume the problem setting in Sec 2. $\forall\phi\in\Phi,\theta\in\Theta, Q^*\in\mathcal{Q}^* := \{\mathcal{X}\times\mathcal{B}\mapsto\mathcal{S}_{\mu,L}^{d\times d}\}$, it holds true that*

$$P\ell_{\phi,\theta}^2 = \varepsilon \implies P\|Q_\theta - Q^*\|_F^2 \leq \sigma_b^{-2}L^4(\sqrt{\varepsilon} + M\cdot \mathit{Cvg}(k,\phi))^2. \tag{4}$$

Proof. We shall prove the same conclusion under a slightly weaker assumption that $P\ell_{\phi,\theta}^2\leq\epsilon$. For any $\boldsymbol{x}\in\mathcal{X}$, $\boldsymbol{b}\in\mathcal{B}$, we have

$$\begin{aligned}
\ell_{\phi,\theta}(\boldsymbol{x}) &\geq \|\texttt{Opt}\left(Q_\theta(\boldsymbol{x}),\boldsymbol{b}\right) - \texttt{Opt}\left(Q^*(\boldsymbol{x}),\boldsymbol{b}\right)\|_2 - \|\texttt{Alg}_\phi^k\left(Q_\theta(\boldsymbol{x}),\boldsymbol{b}\right) - \texttt{Opt}\left(Q_\theta(\boldsymbol{x}),\boldsymbol{b}\right)\|_2 \\
&\geq \|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2 - \mathit{Cvg}(k,\phi)\|\texttt{Opt}\left(Q_\theta(\boldsymbol{x}),\boldsymbol{b}\right)\|_2 && (28)\\
&\geq \|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2 - M\cdot\mathit{Cvg}(k,\phi). && (29)
\end{aligned}$$

Rearranging the terms in the above inequality, we have

$$\|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2 \leq \ell_{\phi,\theta}(\boldsymbol{x}) + M \cdot \boldsymbol{C}vg(k,\phi). \tag{30}$$

By Eq. 22 and the inequality that $\|AB\|_F \leq \|A\|_2\|B\|_F$ for any given $A \in \mathbb{R}^{m\times r}$ and $B \in \mathbb{R}^{r\times n}$, we have that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{b}}\|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2^2 && (31)\\
&= \sigma_b^2\|Q_\theta(\boldsymbol{x})^{-1}(Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x}))Q^*(\boldsymbol{x})^{-1}\|_F^2 && (32)\\
&\geq \sigma_b^2\frac{\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F^2}{\|Q^*(\boldsymbol{x})\|_2^2\|Q_\theta(\boldsymbol{x})\|_2^2} && (33)\\
&\geq \sigma_b^2\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F^2/L^4, && (34)
\end{aligned}$$

which implies that,

$$\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F^2 \leq \sigma_b^{-2}L^4\mathbb{E}_{\boldsymbol{b}}\|Q_\theta(\boldsymbol{x})^{-1}\boldsymbol{b} - Q^*(\boldsymbol{x})^{-1}\boldsymbol{b}\|_2^2. \tag{35}$$

Combining it with Eq. 30 and the fact that $(P\ell_{\phi,\theta})^2 \leq P\ell_{\phi,\theta}^2$, we have

$$\begin{aligned}
P\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F^2 &\leq \sigma_b^{-2}L^4P(\ell_{\phi,\theta} + M\cdot \boldsymbol{C}vg(k,\phi))^2 && (36)\\
&= \sigma_b^{-2}L^4\left(P\ell_{\phi,\theta}^2 + (M\cdot \boldsymbol{C}vg(k,\phi))^2 + 2(M\cdot \boldsymbol{C}vg(k,\phi))P\ell_{\phi,\theta}\right) && (37)\\
&\leq \sigma_b^{-2}L^4\left(\varepsilon + (M\cdot \boldsymbol{C}vg(k,\phi))^2 + 2(M\cdot \boldsymbol{C}vg(k,\phi))\sqrt{\varepsilon}\right) && (38)\\
&= \sigma_b^{-2}L^4\left(\sqrt{\varepsilon} + M\cdot \boldsymbol{C}vg(k,\phi)\right)^2, && (39)
\end{aligned}$$

which completes the proof. □

F. Proof of Generalization Ability

In this section, we shall prove the following result, which is a refined version of Theorem 3.1.

Theorem F.1. *Assume the problem setting in Sec 2 and let $r > 0$. Then for any $t > 0$, with probability at least $1 - e^{-t}$, the empirical Rademacher complexity of $\ell_{\mathcal{F}}^{loc}(r)$ can be bounded by*

$$\begin{aligned} R_n \ell_{\mathcal{F}}^{loc}(r) \leq & \sqrt{2} d n^{-\frac{1}{2}} \mathit{Stab}(k) \left(\sqrt{(\sqrt{r} + M\mathit{Cvg}(k))^2 C_1(n) + C_2(n,t,k,r)} + 4 \right) \\ & + \mathit{Sens}(k) B_\Phi, \end{aligned}$$

where

$$\begin{aligned} C_1(n) &= 216 \sigma_b^{-2} L^4 \log \mathcal{N}(n^{-\frac{1}{2}}, \ell_Q, L_2(P_n)) \\ C_2(n,t,k,r) &= \left(\frac{768 B_Q^2 t}{n} + 720 B_Q \mathbb{E} R_n \ell_Q^{loc}(r_q) \right) \log \mathcal{N}(n^{-\frac{1}{2}}, \ell_Q, L_2(P_n)), \end{aligned}$$

$r_q = \sigma_b^{-2} L^4 (\sqrt{r} + M\mathit{Cvg}(k))^2$, $\ell_Q^{loc}(r_q) = \{\|Q_\theta - Q^*\|_F : \theta \in \Theta, P\|Q_\theta - Q^*\|_F^2 \leq r_q\}$, $B_Q = 2L\sqrt{d}$, and $B_\Phi = \frac{1}{2} \sup_{\phi_1, \phi_2 \in \Phi} \|\phi_1 - \phi_2\|_2$.

Furthermore, for any $t > 0$, the expected Rademacher complexity of $\ell_{\mathcal{F}}^{loc}(r)$ can be bounded by

$$\begin{aligned} \mathbb{E} R_n \ell_{\mathcal{F}}^{loc}(r) \leq & \sqrt{2} d n^{-\frac{1}{2}} \mathit{Stab}(k) \left(\sqrt{(\sqrt{r} + M\mathit{Cvg}(k))^2 \overline{C}_1(n) + \overline{C}_2(n,t)} + \overline{C}_3(n,t) + 4 \right) \\ & + \mathit{Sens}(k) B_\Phi, \end{aligned}$$

where

$$\begin{aligned} \overline{C}_1(n) &= 216 \sigma_b^{-2} L^4 \log \mathcal{N}_Q, \\ \overline{C}_2(n,t) &= \left(1 + 3 B_Q e^{-t} \sqrt{\log \mathcal{N}_Q} + \frac{45}{\sqrt{n}} B_Q \log \mathcal{N}_Q \right) \frac{2880}{\sqrt{n}} B_Q \log \mathcal{N}_Q + t \frac{768 B_Q^2}{n} \log \mathcal{N}_Q, \\ \overline{C}_3(n,t) &= 12 B_Q e^{-t} \sqrt{\log \mathcal{N}_Q} + \frac{360}{\sqrt{n}} B_Q \log \mathcal{N}_Q, \end{aligned}$$

and $\mathcal{N}_Q = \mathcal{N}(n^{-\frac{1}{2}}, \ell_Q, L_\infty)$.

In order to prove Theorem F.1, we first prove the following theorem, which reduces bounding the empirical Rademacher complexity of $\ell_{\mathcal{F}}^{loc}(r)$ to that of $\ell_Q^{loc}(r_q)$, and plays an important role in our complexity analysis.

Theorem F.2. *Assume the problem setting in Sec 2. Then it holds for any $r > 0$ that*

$$R_n \ell_{\mathcal{F}}^{loc}(r) \leq \sqrt{2} d\, \mathit{Stab}(k) R_n \ell_Q^{loc}(r_q) + \mathit{Sens}(k) B_\Phi, \tag{40}$$

with $r_q = \sigma_b^{-2} L^4 (\sqrt{r} + M\mathit{Cvg}(k))^2$, $\ell_Q^{loc}(r_q) = \{\|Q_\theta - Q^*\|_F : \theta \in \Theta, P\|Q_\theta - Q^*\|_F^2 \leq r_q\}$ and $B_\Phi = \frac{1}{2} \sup_{\phi_1, \phi_2 \in \Phi} \|\phi_1 - \phi_2\|_2$.

Proof. Let $k \in \mathbb{N}$ be fixed throughout this proof. We first show that the loss $\ell_{\phi,\theta}$ is $\mathit{Stab}(k)$-Lipschtiz in Q_θ and $\mathit{Sens}(k)$-Lipschitiz in ϕ. For any $(\boldsymbol{x}, \boldsymbol{b}) \in \mathcal{X} \times \mathcal{B}$, by using the triangle inequality and the definitions of $\mathit{Stab}(k, \phi')$ and $\mathit{Sens}(k)$, we can obtain the following estimate of the loss:

$$\begin{aligned} & |\ell_{\phi,\theta}(\boldsymbol{x}) - \ell_{\phi',\theta'}(\boldsymbol{x})| \\ & = \big| \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})\|_2 - \|\texttt{Alg}_{\phi'}^k(Q_{\theta'}(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})\|_2 \big| \\ & \leq \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Alg}_{\phi'}^k(Q_{\theta'}(\boldsymbol{x}), \boldsymbol{b})\|_2 \\ & \leq \|\texttt{Alg}_{\phi'}^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Alg}_{\phi'}^k(Q_{\theta'}(\boldsymbol{x}), \boldsymbol{b})\|_2 + \|\texttt{Alg}_\phi^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b}) - \texttt{Alg}_{\phi'}^k(Q_\theta(\boldsymbol{x}), \boldsymbol{b})\|_2 \\ & \leq \mathit{Stab}(k, \phi') \|Q_\theta(\boldsymbol{x}) - Q_{\theta'}(\boldsymbol{x})\|_2 + \mathit{Sens}(k) \|\phi - \phi'\|_2 \\ & \leq \mathit{Stab}(k) \|Q_\theta(\boldsymbol{x}) - Q_{\theta'}(\boldsymbol{x})\|_2 + \mathit{Sens}(k) \|\phi - \phi'\|_2. \end{aligned} \tag{41}$$

where we write $\mathit{Stab}(k) = \sup_{\phi \in \Phi} \mathit{Stab}(k, \phi)$ for each $k \in \mathbb{N}$.

We then establish a vector contraction inequality, which is a modified version of Corollary 4 in [20] and Lemma 5 in [21]. Note that the empirical Rademacher complexity of $\ell_{\mathcal{F}}^{loc}$ can be written as:

$$R_n \ell_{\mathcal{F}}^{loc}(r) = \frac{1}{n} \mathbb{E}_\sigma \sup_{\phi,\theta} \sum_{i=1}^{n} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) = \frac{1}{n} \mathbb{E}_{\sigma_{1:n-1}} \mathbb{E}_{\sigma_n} \sup_{\phi,\theta} \sum_{i=1}^{n-1} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) + \sigma_n \ell_{\phi,\theta}(\boldsymbol{x}_n), \tag{42}$$

where the supremum is taken over the parameter space $\left\{(\phi, \theta) : \phi \in \Phi, \theta \in \Theta, P\ell_{\phi,\theta}^2 \le r\right\}$.

Let $U_{n-1}(\phi, \theta) = \sum_{i=1}^{n-1} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i)$ for each (ϕ, θ). We now assume without loss of generality that the supremum can be attained and let

$$\begin{aligned} \phi_1, \theta_1 &= \arg\sup_{\phi,\theta} \big(U_{n-1}(\phi,\theta) + \ell_{\phi,\theta}(\boldsymbol{x}_n)\big), \\ \phi_2, \theta_2 &= \arg\sup_{\phi,\theta} \big(U_{n-1}(\phi,\theta) - \ell_{\phi,\theta}(\boldsymbol{x}_n)\big), \end{aligned}$$

since otherwise we can consider (ϕ_1, θ_1) and (ϕ_2, θ_2) that are ϵ-close to the suprema for any $\epsilon > 0$ and conclude the same result. Then we can deduce from Eq. 41 that

$$\begin{aligned} &\mathbb{E}_{\sigma_n} \sup_{\phi,\theta} \sum_{i=1}^{n-1} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) + \sigma_n \ell_{\phi,\theta}(\boldsymbol{x}_n) \\ &= \frac{1}{2}\big(U_{n-1}(\phi_1,\theta_1) + \ell_{\phi_1,\theta_1}(\boldsymbol{x}_n) + U_{n-1}(\phi_2,\theta_2) - \ell_{\phi_2,\theta_2}(\boldsymbol{x}_n)\big) \\ &= \frac{1}{2}\big(U_{n-1}(\phi_1,\theta_1) + U_{n-1}(\phi_2,\theta_2) + (\ell_{\phi_1,\theta_1}(\boldsymbol{x}_n) - \ell_{\phi_2,\theta_2}(\boldsymbol{x}_n))\big) \\ &\le \frac{1}{2}\big(U_{n-1}(\phi_1,\theta_1) + U_{n-1}(\phi_2,\theta_2)\big) + \frac{1}{2}\big(\mathit{Stab}(k)\|Q_{\theta_1}(\boldsymbol{x}_n) - Q_{\theta_2}(\boldsymbol{x}_n)\|_2 + \mathit{Sens}(k)\|\phi_1 - \phi_2\|_2\big) \\ &\le \frac{1}{2}\big(U_{n-1}(\phi_1,\theta_1) + U_{n-1}(\phi_2,\theta_2)\big) + \frac{1}{2}\mathit{Stab}(k)\|Q_{\theta_1}(\boldsymbol{x}_n) - Q_{\theta_2}(\boldsymbol{x}_n)\|_F + \mathit{Sens}(k) B_\Phi, \end{aligned}$$

where $B_\Phi = \frac{1}{2}\sup_{\phi_1,\phi_2 \in \Phi} \|\phi_1 - \phi_2\|_2$.

For each $\boldsymbol{x} \in \mathcal{X}$, $\theta \in \Theta$ and $1 \le j,k \le d$, let $Q_\theta^{j,k}(\boldsymbol{x})$ be the j,k-th entry of the matrix $Q_\theta(\boldsymbol{x})$. The the Khintchine-Kahane inequality (see e.g. [20]) gives us that

$$\mathbb{E}_{\sigma_n} \sup_{\phi,\theta} \sum_{i=1}^{n} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) \le \frac{1}{2}\left(U_{n-1}(\phi_1,\theta_1) + U_{n-1}(\phi_2,\theta_2)\right) + \mathit{Sens}(k) B_\Phi \tag{43}$$

$$+ \frac{1}{2}\mathit{Stab}(k)\sqrt{2}\,\mathbb{E}_{\boldsymbol{\epsilon}_n}\left| \sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta_1}^{j,k}(\boldsymbol{x}_n) - Q_{\theta_2}^{j,k}(\boldsymbol{x}_n)\right) \right|, \tag{44}$$

where $\boldsymbol{\epsilon}_n = (\epsilon_n^{j,k})_{j,k=1}^n$ are independent Rademacher variables. Hence, if we denote by $s(\boldsymbol{\epsilon}_n)$ the sign of

$\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta_1}^{j,k}(\boldsymbol{x}_n) - Q_{\theta_2}^{j,k}(\boldsymbol{x}_n)\right)$ and by $Q^{*j,k}(\boldsymbol{x})$ be the j,k-th entry of the matrix $Q^*(\boldsymbol{x})$, then we can obtain that

$$
\begin{aligned}
&\mathbb{E}_{\sigma_n} \sup_{\phi,\theta} \sum_{i=1}^{n} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) \\
&\le \mathbb{E}_{\boldsymbol{\epsilon}_n} \frac{1}{2}\Bigg[\Bigg(U_{n-1}(\phi_1,\theta_1) + \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} Q_{\theta_1}^{j,k}(\boldsymbol{x}_n)\Bigg) \\
&\quad + \Bigg(U_{n-1}(\phi_2,\theta_2) - \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} Q_{\theta_2}^{j,k}(\boldsymbol{x}_n)\Bigg)\Bigg] + \mathit{Sens}(k)B_\Phi \\
&= \mathbb{E}_{\boldsymbol{\epsilon}_n} \frac{1}{2}\Bigg[\Bigg(U_{n-1}(\phi_1,\theta_1) + \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta_1}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg) \\
&\quad + \Bigg(U_{n-1}(\phi_2,\theta_2) - \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta_2}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg)\Bigg] + \mathit{Sens}(k)B_\Phi.
\end{aligned}
$$

Then by taking the supremum over (ϕ,θ) and using the fact that σ_n is an independent Rademacher variable, we can deduce that

$$
\begin{aligned}
&\mathbb{E}_{\sigma_n} \sup_{\phi,\theta} \sum_{i=1}^{n} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) \\
&\le \mathbb{E}_{\boldsymbol{\epsilon}_n} \frac{1}{2}\Bigg[\sup_{\phi,\theta}\Bigg(U_{n-1}(\phi,\theta) + \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg) \\
&\quad + \sup_{\phi,\theta}\Bigg(U_{n-1}(\phi,\theta) - \mathit{Stab}(k)\sqrt{2}s(\boldsymbol{\epsilon}_n)\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg)\Bigg] + \mathit{Sens}(k)B_\Phi \\
&= \mathbb{E}_{\boldsymbol{\epsilon}_n}\mathbb{E}_{\sigma_n}\Bigg[\sup_{\phi,\theta}\Bigg(U_{n-1}(\phi,\theta) + \mathit{Stab}(k)\sqrt{2}\sigma_n\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg)\Bigg] + \mathit{Sens}(k)B_\Phi \\
&= \mathbb{E}_{\boldsymbol{\epsilon}_n}\Bigg[\sup_{\phi,\theta}\Bigg(U_{n-1}(\phi,\theta) + \mathit{Stab}(k)\sqrt{2}\sum_{j,k} \epsilon_n^{j,k} \left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\Bigg)\Bigg] + \mathit{Sens}(k)B_\Phi,
\end{aligned}
$$

where we have used the fact that $\sum_{j,k} \epsilon_n^{j,k}\left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)$ is a symmetric random variable in the last line.

By proceeding in the same way for all other $\sigma_{n-1},\cdots,\sigma_1$, we can obtain the following vector-contraction inequality:

$$
\begin{aligned}
\mathbb{E}_{\sigma} \sup_{\phi,\theta} \sum_{i=1}^{n} \sigma_i \ell_{\phi,\theta}(\boldsymbol{x}_i) \le & \sqrt{2}\mathit{Stab}(k)\mathbb{E}_{\boldsymbol{\epsilon}_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sum_{j,k} \epsilon_i^{j,k}\left(Q_{\theta}^{j,k}(\boldsymbol{x}_n) - Q^{*j,k}(\boldsymbol{x}_n)\right)\right] \\
& + n\mathit{Sens}(k)B_\Phi.
\end{aligned}
\tag{45}
$$

The first term on the right-hand side can be bounded by using the Cauchy-Schwarz inequality as follows:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\epsilon}_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sum_{j,k}\epsilon_i^{j,k}\left(Q_\theta^{j,k}(\boldsymbol{x}_n)-Q^{*j,k}(\boldsymbol{x}_n)\right)\right]\\
&=\mathbb{E}_{\sigma_{1:n}}\mathbb{E}_{\boldsymbol{\epsilon}_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sigma_i\sum_{j,k}\epsilon_i^{j,k}\left(Q_\theta^{j,k}(\boldsymbol{x}_n)-Q^{*j,k}(\boldsymbol{x}_n)\right)\right]\\
&\leq\mathbb{E}_{\sigma_{1:n}}\mathbb{E}_{\boldsymbol{\epsilon}_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sigma_i\sqrt{\sum_{j,k}(\epsilon_i^{j,k})^2}\sqrt{\sum_{j,k}|Q_\theta^{j,k}(\boldsymbol{x}_n)-Q^{*j,k}(\boldsymbol{x}_n)|^2}\right]\\
&=\mathbb{E}_{\sigma_{1:n}}\mathbb{E}_{\boldsymbol{\epsilon}_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sigma_i d\|Q_\theta(\boldsymbol{x}_n)-Q^*(\boldsymbol{x}_n)\|_F\right]\\
&=d\mathbb{E}_{\sigma_{1:n}}\left[\sup_{\theta}\sum_{i=1}^{n}\sigma_i\|Q_\theta(\boldsymbol{x}_n)-Q^*(\boldsymbol{x}_n)\|_F\right].
\end{aligned}\tag{46}$$

Therefore, bounding the Rademacher complexity of $\ell_{\mathcal{F}}^{loc}(r)$ reduces to bounding the Rademacher complexity of the space of functions $\|Q_\theta - Q^*\|_F$. Recall that the supremum is taken over the parameter space where $(\phi,\theta)\in\Phi\times\Theta$ satisfies $P\ell_{\phi,\theta}^2\leq r$. Note that Lemma A.2 implies that,

$$P\|Q_\theta-Q^*\|_F^2\leq r_q:=\sigma_b^{-2}L^4\left(\sqrt{\varepsilon}+M\cdot Cvg(k,\phi)\right)^2.\tag{47}$$

Hence, by defining the following function space:

$$\ell_{\mathcal{Q}}^{loc}(r_q):=\left\{\|Q_\theta-Q^*\|_F:\theta\in\Theta,P\|Q_\theta-Q^*\|_F^2\leq r_q\right\},\tag{48}$$

we can conclude the desired relationship between $R_n\ell_{\mathcal{F}}^{loc}(r)$ and $R_n\ell_{\mathcal{Q}}^{loc}(r_q)$ from the inequalities Eq. 45 and Eq. 46.

□

With Theorem F.2 in hand, we see that, for each $r>0$, in order to obtain the upper bounds of $R_n\ell_{\mathcal{F}}^{loc}(r)$ in Theorem F.1, it suffices to estimate $R_n\ell_{\mathcal{Q}}^{loc}(r_q)$, i.e., the Rademacher complexity of the function space $\ell_{\mathcal{Q}}^{loc}(r_q)$.

The following theorem summarizes the estimates for the empirical and expected Rademacher complexity of the local class $\ell_{\mathcal{Q}}^{loc}$, which will be established in Propositions F.1 and F.2, respectively.

Recall that, for any given $\epsilon>0$, a class of functions $\mathcal{F}$ and pseudometric $\|\cdot\|$, the covering number $\mathcal{N}(\epsilon,\mathcal{F},\|\cdot\|)$ is defined as the cardinality of the smallest subset $\hat{\mathcal{F}}$ of $\mathcal{F}$ for which every element of $\mathcal{F}$ is within the ϵ-neighbourhood of some element of $\hat{\mathcal{F}}$ with respect to the pseudometric $\|\cdot\|$.

Theorem F.3. *Assume the problem setting in Sec 2. Let $r>0$, $r_q=\sigma_b^{-2}L^4(\sqrt{r}+MCvg(k))^2$ and $\ell_{\mathcal{Q}}^{loc}(r_q)=\{\|Q_\theta-Q^*\|_F:\theta\in\Theta,P\|Q_\theta-Q^*\|_F^2\leq r_q\}$. Then for all $t>0$, we have with probability at least $1-e^{-t}$ that*

$$R_n\ell_{\mathcal{Q}}^{loc}(r_q)\leq n^{-\frac{1}{2}}\left[\left(C_1(n)(\sqrt{r}+MCvg(k))^2+C_2(n,t,k,r)\right)^{\frac{1}{2}}+4\right],\tag{49}$$

where

$$\begin{aligned}
C_1(n)&=216\sigma_b^{-2}L^4\log\mathcal{N}\big(n^{-\frac{1}{2}},\ell_{\mathcal{Q}},L_2(P_n)\big),\\
C_2(n,t,k,r)&=\left(\frac{768B_Q^2t}{n}+720B_Q\mathbb{E}R_n\ell_{\mathcal{Q}}^{loc}(r_q)\right)\log\mathcal{N}\big(n^{-\frac{1}{2}},\ell_{\mathcal{Q}},L_2(P_n)\big),
\end{aligned}$$

and $B_Q=2L\sqrt{d}$.

Moreover, for all $t > 0$, we have that

$$\mathbb{E}R_n \ell_{\mathcal{Q}}^{loc}(r_q) \leq n^{-\frac{1}{2}}\left[\left(\overline{C}_1(n)(\sqrt{r} + MCvg(k))^2 + \overline{C}_2(n,t)\right)^{\frac{1}{2}} + \overline{C}_3(n,t) + 4\right], \tag{50}$$

where

$$\begin{aligned}
\overline{C}_1(n) &= 216\sigma_b^{-2}L^4 \log \mathcal{N}_Q, \\
\overline{C}_2(n,t) &= \left(1 + 3B_Q e^{-t}\sqrt{\log \mathcal{N}_Q} + \frac{45}{\sqrt{n}}B_Q \log \mathcal{N}_Q\right)\frac{2880}{\sqrt{n}}B_Q \log \mathcal{N}_Q + t\frac{768B_Q^2}{n}\log \mathcal{N}_Q, \\
\overline{C}_3(n,t) &= 12B_Q e^{-t}\sqrt{\log \mathcal{N}_Q} + \frac{360}{\sqrt{n}}B_Q \log \mathcal{N}_Q
\end{aligned}$$

and $\mathcal{N}_Q = \mathcal{N}(n^{-\frac{1}{2}}, \ell_{\mathcal{Q}}, L_\infty)$.

We first establish the estimate for the empirical Rademacher complexity $R_n \ell_{\mathcal{Q}}^{loc}(r_q)$, i.e., Eq. 49 in Theorem F.3.

Proposition F.1. *Assume the problem setting in Sec 2. Let $B_Q = \sup_{(\theta, \boldsymbol{x}) \in \Theta \times \mathcal{X}} \|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F$, and for each $r > 0$ let r_q and $\ell_{\mathcal{Q}}^{loc}(r_q)$ be defined as in Theorem F.2. Then we have that*

$$R_n \ell_{\mathcal{Q}}^{loc}(r_q) \leq \tfrac{4}{\sqrt{n}}\left(1 + 3B_Q\sqrt{\log \mathcal{N}\left(\tfrac{1}{\sqrt{n}}, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n)\right)}\right). \tag{51}$$

Moreover, for all $t > 0$, it holds with probability at least $1 - e^{-t}$ that

$$R_n \ell_{\mathcal{Q}}^{loc}(r_q) \leq \tfrac{4}{\sqrt{n}}\left(1 + 3C(r_q, t)\sqrt{\log \mathcal{N}\left(\tfrac{1}{\sqrt{n}}, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n)\right)}\right), \tag{52}$$

with the constant $C(r_q, t) = \left(\frac{3r_q}{2} + \frac{16B_Q^2 t}{3n} + 5B_Q \mathbb{E}R_n \ell_{\mathcal{Q}}^{loc}(r_q)\right)^{1/2}$.

Proof. The classical Dudley's entropy integral bound for the empirical Rademacher complexity gives us that

$$R_n \ell_{\mathcal{Q}}^{loc}(r_q) \leq \inf_{\alpha > 0}\left(4\alpha + \frac{12}{\sqrt{n}}\int_\alpha^\infty \sqrt{\log \mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n))}\, d\epsilon\right). \tag{53}$$

Observe that all functions in $\ell_{\mathcal{Q}}^{loc}(r_q)$ take value in $[0, B_Q]$, which implies for all $\epsilon \geq B_Q$ that, $\mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n)) \leq \mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_\infty(P_n)) = 1$ and consequently the integrand in Eq. 53 vanishes on $[B_Q, \infty)$. Hence we have that

$$\begin{aligned}
R_n \ell_{\mathcal{Q}}^{loc}(r_q) &\leq \inf_{\alpha > 0}\left(4\alpha + \frac{12}{\sqrt{n}}\int_\alpha^{B_Q} \sqrt{\log \mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n))}\, d\epsilon\right) \\
&\leq \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}\int_{\frac{1}{\sqrt{n}}}^{B_Q} \sqrt{\log \mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n))}\, d\epsilon \\
&\leq \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}B_Q\sqrt{\log \mathcal{N}\left(\frac{1}{\sqrt{n}}, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n)\right)},
\end{aligned}$$

where we used the fact that $\mathcal{N}(\epsilon, \ell_{\mathcal{Q}}^{loc}(r_q), L_2(P_n))$ is decreasing in terms of ϵ for the last inequality. This proves the estimate Eq. 51.

In order to establish the estimate Eq. 52, we shall bound the empirical error $P_n\|Q_\theta - Q^*\|_F^2$ with high probability. Let us consider the class of functions $\ell_{\mathcal{Q}^2}^{loc}(r_q) = \{\|Q_\theta - Q^*\|_F^2 : \theta \in \Theta, P\|Q_\theta - Q^*\|_F^2 \leq r_q\}$, whose element takes values in $[0, B_Q^2]$. Moreover, we see it holds for all $\|Q_\theta - Q^*\|_F^2 \in \ell_{\mathcal{Q}^2}^{loc}(r_q)$ that $P\|Q_\theta - Q^*\|_F^4 \leq B_Q^2 P\|Q_\theta - Q^*\|_F^2 \leq B_Q^2 r_q$. Hence, by applying Theorem 2.1 in [15] (with $\mathcal{F} = \ell_{\mathcal{Q}^2}^{loc}(r_q)$, $a = 0$, $b = B_Q^2$, $\alpha = 1/4$ and $r = B_Q^2 r_q$) and the Cauchy-Schwarz inequality, we can deduce that, for each $t > 0$, it holds with probability at least $1 - e^{-t}$ that

$$\begin{aligned} P_n\|Q_\theta - Q^*\|_F^2 &\le P\|Q_\theta - Q^*\|_F^2 + \frac{5}{2}\mathbb{E}R_n\ell_{Q^2}^{loc}(r_q) + \sqrt{\frac{2B_Q^2 r_q t}{n}} + B_Q^2\frac{13t}{3n} \\ &\le r_q + \frac{5}{2}\mathbb{E}R_n\ell_{Q^2}^{loc}(r_q) + \frac{r_q}{2} + \frac{B_Q^2 t}{n} + B_Q^2\frac{13t}{3n} \\ &\le \frac{3r_q}{2} + 5B_Q\mathbb{E}R_n\ell_Q^{loc}(r_q) + \frac{16B_Q^2 t}{3n}. \end{aligned}$$

Consequently, we see it holds with probability at least $1 - e^{-t}$ that, $\mathcal{N}(\epsilon, \ell_Q^{loc}(r_q), L_2(P_n)) = 1$ for all $\epsilon \ge C(r_q, t)$, with the constant $C(r_q, t)$ defined as in the statement of Proposition F.1. Substituting this fact into the integral bound Eq. 53 and following the same argument as above, we can conclude Eq. 52 with probability at least $1 - e^{-t}$. □

Now we proceed to prove the estimate of the expected Rademacher complexity $\mathbb{E}R_n\ell_Q^{loc}(r_q)$, i.e., Eq. 50 in Theorem F.3.

Proposition F.2. *Assume the same setting as in Proposition F.1. Then it holds for any $r, t > 0$ that*

$$\mathbb{E}R_n\ell_Q^{loc}(r_q) \le n^{-\frac{1}{2}}\left[\left(C_1(n,t)(\sqrt{r} + MCvg(k))^2 + C_2(n,t)\right)^{\frac{1}{2}} + C_3(n,t) + 4\right], \tag{54}$$

where $C_1(n,t)$, $C_2(n,t)$ and $C_3(n,t)$ the constants defined as in Eq. 57, Eq. 58 and Eq. 59, respectively.

Proof. Let $r, t > 0$ be fixed throughout this proof. Since it holds for all $\epsilon > 0$ and $n \in \mathbb{N}$ that $\mathcal{N}(\epsilon, \ell_Q^{loc}(r_q), L_2(P_n)) \le \mathcal{N}(\epsilon, \ell_Q, L_\infty)$, we can deduce from Proposition F.1 that

$$\mathbb{E}R_n\ell_Q^{loc}(r_q) \le \tfrac{4}{\sqrt{n}}\left(1 + 3\big[C(r_q,t)(1 - e^{-t}) + B_Q e^{-t}\big]\sqrt{\log\mathcal{N}\big(\tfrac{1}{\sqrt{n}}, \ell_Q, L_\infty\big)}\right), \tag{55}$$

with the constants B_Q and $C(r_q, t)$ defined as in the statement of Proposition F.1.

The above estimate gives an implicit upper bound of $\mathbb{E}R_n\ell_Q^{loc}(r_q)$ since $C(r_q, t)$ also involves $\mathbb{E}R_n\ell_Q^{loc}(r_q)$. Now we shall introduce the notation $\mathcal{N}_Q^n = \mathcal{N}(\frac{1}{\sqrt{n}}, \ell_Q, L_\infty)$ and derive an explicit upper bound of $\mathbb{E}R_n\ell_Q^{loc}(r_q)$. By rearranging the terms in Eq. 55 and using the definition of $C(r_q, t)$, we can obtain that

$$\begin{aligned} &\frac{\sqrt{n}}{4}\mathbb{E}R_n\ell_Q^{loc}(r_q) - 1 - 3B_Q e^{-t}\sqrt{\log\mathcal{N}_Q^n} \\ &\le 3(1 - e^{-t})\sqrt{\left(\frac{3r_q}{2} + \frac{16B_Q^2 t}{3n} + 5B_Q\mathbb{E}R_n\ell_Q^{loc}(r_q)\right)\log\mathcal{N}_Q^n}. \end{aligned} \tag{56}$$

We shall assume without loss of generality that $\mathbb{E}R_n\ell_Q^{loc}(r_q) \ge \frac{4}{\sqrt{n}}\left(1 + 3B_Q e^{-t}\sqrt{\log\mathcal{N}_Q^n}\right)$, since otherwise we have a trivial estimate that $\mathbb{E}R_n\ell_Q^{loc}(r_q) \le 4n^{-\frac{1}{2}}A_1$, with $A_1 = 1 + 3B_Q e^{-t}\sqrt{\log\mathcal{N}_Q^n}$. Then by squaring both sides of Eq. 56 and rearranging the terms, we get that

$$\begin{aligned} &\frac{n}{16}(\mathbb{E}R_n\ell_Q^{loc}(r_q))^2 - \left(\frac{\sqrt{n}}{2}A_1 + 45(1 - e^{-t})^2 B_Q\log\mathcal{N}_Q^n\right)\mathbb{E}R_n\ell_Q^{loc}(r_q) \\ &+ A_1^2 - 9(1 - e^{-t})^2 A_2\log\mathcal{N}_Q^n \le 0, \end{aligned}$$

with the constant $A_2 = \frac{3r_q}{2} + \frac{16B_Q^2 t}{3n}$. This implies that

$$\begin{aligned}\mathbb{E}R_n\ell_{\mathcal{Q}}^{loc}(r_q) \le \frac{8}{n}\Bigg[&\frac{\sqrt{n}A_1}{2} + 45(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n + \Bigg(\big[\frac{\sqrt{n}A_1}{2} + 45(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n\big]^2 \\ &- \frac{n}{4}\big[A_1^2 - 9(1-e^{-t})^2 A_2 \log\mathcal{N}_Q^n\big]\Bigg)^{\frac{1}{2}}\Bigg] \\ = n^{-\frac{1}{2}}\Bigg[&4A_1 + \frac{360}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n + \Bigg(\big[4A_1 + \frac{360}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n\big]^2 \\ &- 16\big[A_1^2 - 9(1-e^{-t})^2 A_2 \log\mathcal{N}_Q^n\big]\Bigg)^{\frac{1}{2}}\Bigg].\end{aligned}$$

Hence, for each $t > 0$, by introducing the following constants

$$C_1(n,t) = 216(1-e^{-t})^2\sigma_b^{-2}L^4\log\mathcal{N}_Q^n, \tag{57}$$

$$\begin{aligned}C_2(n,t) &= \big[4A_1 + \frac{360}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n\big]^2 - 16A_1^2 + t(1-e^{-t})^2\frac{768B_Q^2}{n}\log\mathcal{N}_Q^n \\ &= \Bigg(1 + 3B_Q e^{-t}\sqrt{\log\mathcal{N}_Q^n} + \frac{45}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n\Bigg)\frac{2880}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n \\ &\quad + t(1-e^{-t})^2\frac{768B_Q^2}{n}\log\mathcal{N}_Q^n,\end{aligned} \tag{58}$$

$$C_3(n,t) = 12B_Q e^{-t}\sqrt{\log\mathcal{N}_Q^n} + \frac{360}{\sqrt{n}}(1-e^{-t})^2 B_Q \log\mathcal{N}_Q^n, \tag{59}$$

with $B_Q = \sup_{(\theta,\boldsymbol{x})\in\Theta\times\mathcal{X}}\|Q_\theta(\boldsymbol{x}) - Q^*(\boldsymbol{x})\|_F \le 2\sqrt{d}L$ and $\mathcal{N}_Q^n = \mathcal{N}(\frac{1}{\sqrt{n}}, \ell_{\mathcal{Q}}, L_\infty)$, we can deduce that

$$\mathbb{E}R_n\ell_{\mathcal{Q}}^{loc}(r_q) \le n^{-\frac{1}{2}}\Bigg[\Bigg(C_1(n,t)(\sqrt{r} + MCvg(k))^2 + C_2(n,t)\Bigg)^{\frac{1}{2}} + C_3(n,t) + 4\Bigg].$$

□

G. Poof of Algorithm properties of RNN

We denote by $\texttt{RNN}_\phi^k$ a recurrent neural network that has k unrolled RNN cells and view it as a neural algorithm. It has been proposed in [18] to use RNN to learn an optimization algorithm where the update steps in each iteration are given by the operations in an RNN cell

$$\boldsymbol{y}_{k+1} \leftarrow \texttt{RNNcell}\,(Q, \boldsymbol{b}, \boldsymbol{y}_k) := V\sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_k + W_2^1\boldsymbol{g}_k\right)\right)\right). \tag{60}$$

In the above equation, we take a specific example where the $\texttt{RNNcell}$ is a multi-layer perception (MLP) with activations $\sigma = \text{RELU}$ that takes the current iterate $\boldsymbol{y}_k$ and the gradient $\boldsymbol{g}_k = Q\boldsymbol{y}_k + \boldsymbol{b}$ as inputs.

(I) Stable Region. First, we show that when the parameters satisfy $c_\phi := \sup_Q \|V\|_2\|W_1^1 + W_2^1Q\|_2\prod_{l=2}^L\|W^l\|_2 < 1$, the operations in $\texttt{RNNcell}$ are strictly contractive, i.e., $\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|_2 \leq c_\phi\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|_2$.

Proof. By definition,

$$\begin{aligned}\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|_2 &= \|V\sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_k + W_2^1\boldsymbol{g}_k\right)\right)\right)\\ &\qquad - V\sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_{k-1} + W_2^1\boldsymbol{g}_{k-1}\right)\right)\right)\|_2\\ &\leq \|V\|_2\|\sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_k + W_2^1\boldsymbol{g}_k\right)\right)\right)\\ &\qquad - \sigma\left(W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_{k-1} + W_2^1\boldsymbol{g}_{k-1}\right)\right)\right)\|_2\end{aligned}$$

Since the activation function $\sigma = \text{RELU}$ satisfies the inequality that $\|\sigma(\boldsymbol{x}) - \sigma(\boldsymbol{x}')\|_2 \leq \|\boldsymbol{x} - \boldsymbol{x}'\|_2$ for any $\boldsymbol{x}, \boldsymbol{x}'$, we have

$$\begin{aligned}\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|_2 \leq \|V\|_2\|W^L\sigma&\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_k + W_2^1\boldsymbol{g}_k\right)\right)\\ &- W^L\sigma\left(W^{L-1}\cdots W^2\sigma\left(W_1^1\boldsymbol{y}_{k-1} + W_2^1\boldsymbol{g}_{k-1}\right)\right)\|_2.\end{aligned}$$

Similarly, we can obtain

$$\begin{aligned}&\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|_2\\ &\leq \|V\|_2\|W^L\|_2\cdots\|W^2\|_2\|\left(W_1^1\boldsymbol{y}_k + W_2^1\boldsymbol{g}_k\right) - \left(W_1^1\boldsymbol{y}_{k-1} + W_2^1\boldsymbol{g}_{k-1}\right)\|_2\\ &= \|V\|_2\|W^L\|_2\cdots\|W^2\|_2\|(W_1^1 + QW_2^1)(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})\|_2\\ &\leq \|V\|_2\|W^L\|_2\cdots\|W^2\|_2\|W_1^1 + QW_2^1\|_2\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|_2\\ &\leq c_\phi\|\boldsymbol{y}_k - \boldsymbol{y}_{k-1}\|_2.\end{aligned}$$

Therefore, if $c_\phi < 1$, then the operation is strictly contractive. □

(II) Stability. We shall show the neural algorithm $\texttt{RNN}_\phi^k$ has a stability constant *Stab*$(k, \phi) = \mathcal{O}(1 - c_\phi^k)$ (see the definition of stability in Sec 3).

Proof. Let us consider two quadratic problems induced by $(Q, \boldsymbol{b})$ and $(Q', \boldsymbol{b}')$, and denote the corresponding outputs of $\texttt{RNN}_\phi^k$ as $\boldsymbol{y}_k = \texttt{RNN}_\phi^k(Q, \boldsymbol{b})$ and $\boldsymbol{y}_k' = \texttt{RNN}_\phi^k(Q', \boldsymbol{b}')$.

Denote $c_\phi^Q = \|V\|_2\|W_1^1 + W_2^1Q\|_2\prod_{l=2}^L\|W^l\|_2$, $c_\phi^{Q'} = \|V\|_2\|W_1^1 + W_2^1Q'\|_2\prod_{l=2}^L\|W^l\|_2$, and $\hat{c}_\phi := \|V\|_2\|W_2^1\|_2\prod_{l=2}^L\|W^l\|_2$. First, we see that

$$\begin{aligned}\|\boldsymbol{y}_k\|_2 \leq c_\phi^Q\|\boldsymbol{y}_{k-1}\|_2 + \hat{c}_\phi\|\boldsymbol{b}\|_2 &\leq (c_\phi^Q)^k\|\boldsymbol{y}_0\|_2 + \hat{c}_\phi\|\boldsymbol{b}\|_2\sum_{i=1}^k(c_\phi^Q)^{i-1}\\ &= \frac{\hat{c}_\phi\|\boldsymbol{b}\|_2(1 - (c_\phi^Q)^k)}{1 - c_\phi^Q} \leq \frac{\hat{c}_\phi\|\boldsymbol{b}\|_2}{1 - c_\phi^Q}.\end{aligned} \tag{61}$$

Similar conclusion holds for $\boldsymbol{y}'_k$. Then, by following a similar argument as that for the proof of the stable region, we can deduce from $\boldsymbol{y}_0 = \boldsymbol{y}'_0$ that

$$
\begin{aligned}
&\|\boldsymbol{y}_k - \boldsymbol{y}'_k\|_2 \\
&\leq \|V\|_2 \|W^L\|_2 \cdots \|W^2\|_2 \|(W_1^1 + W_1^2 Q)\boldsymbol{y}_{k-1} - (W_1^1 + W_1^2 Q')\boldsymbol{y}'_{k-1} + W_1^2(\boldsymbol{b} - \boldsymbol{b}')\|_2 \\
&\leq \|V\|_2 \|W^L\|_2 \cdots \|W^2\|_2 (\|W_1^1 + W_1^2 Q\|_2 \|\boldsymbol{y}_{k-1} - \boldsymbol{y}'_{k-1}\|_2 + \|Q - Q'\|_2 \|W_1^2\|_2 \|\boldsymbol{y}'_{k-1}\|_2 \\
&+ \|W_1^2\|_2 \|(\boldsymbol{b} - \boldsymbol{b}')\|_2) \\
&\leq c_\phi^Q \|\boldsymbol{y}_{k-1} - \boldsymbol{y}'_{k-1}\|_2 + \hat{c}_\phi \|Q - Q'\|_2 \frac{\hat{c}_\phi \|\boldsymbol{b}'\|_2}{1 - c_\phi^{Q'}} + \hat{c}_\phi \|\boldsymbol{b} - \boldsymbol{b}'\|_2 \\
&\leq (c_\phi^Q)^k \|\boldsymbol{y}_0 - \boldsymbol{y}'_0\|_2 + \left(\frac{\hat{c}_\phi^2 \|\boldsymbol{b}'\|_2}{1 - c_\phi^{Q'}} \|Q - Q'\|_2 + \hat{c}_\phi \|\boldsymbol{b} - \boldsymbol{b}'\|_2 \right) \sum_{i=1}^k (c_\phi^Q)^{i-1} \\
&= \frac{\hat{c}_\phi^2 \|\boldsymbol{b}'\|_2}{1 - c_\phi^{Q'}} \frac{1 - (c_\phi^Q)^k}{1 - c_\phi^Q} \|Q - Q'\|_2 + \hat{c}_\phi \frac{1 - (c_\phi^Q)^k}{1 - c_\phi^Q} \|\boldsymbol{b} - \boldsymbol{b}'\|_2.
\end{aligned}
$$

Therefore, the stability constant is of the magnitude $\mathcal{O}(1 - c_\phi^k)$. □

(III) Sensitivity. We now proceed to analyze the sensitivity of the neural algorithm $\texttt{RNN}_\phi^k$ as defined in Sec 3. Note that the strong non-linearity in the RNN cell and the high-dimensionality of the parameter space significantly complicate the analysis of the Lipschitz dependence of $\texttt{RNN}_\phi^k$ with respect to its parameter $\phi = \{W_1^1, W_1^1, W^2, \ldots, W^L, V\}$. To simplify our presentation, we shall assume the parameter ϕ are constrained in a compact subset Φ of the stable region, and show the neural algorithm $\texttt{RNN}_\phi^k$ has a sensitivity *Sens*$(k) = \mathcal{O}(1 - (\inf_{\phi \in \Phi} c_\phi)^k)$. A rigorous sensitivity analysis of RNN with general weights is out of the scope of this paper.

Proof. Let the range of parameters Φ is a compact subset of the stable region, such that for all $\phi \in \Phi$, $c_\phi := \sup_Q \|V\|_2 \|W_1^1 + W_2^1 Q\|_2 \prod_{l=2}^L \|W^l\|_2 \leq c_0 < 1$ for some constant c_0. Let $\phi, \phi' \in \Phi$ be two given sets of parameters. For each $k \in \mathbb{N}$, we denote $\boldsymbol{y}_k = \texttt{RNN}_\phi^k(Q, \boldsymbol{b})$ and $\boldsymbol{y}'_k = \texttt{RNN}_{\phi'}^k(Q, \boldsymbol{b})$ the outputs corresponding to the parameters ϕ and ϕ', respectively. Then we have that

$$
\begin{aligned}
\|\boldsymbol{y}_k - \boldsymbol{y}'_k\|_2 &= \|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}_{k-1}) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1})\|_2 \\
&\leq \|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1}) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1})\|_2 \\
&\quad + \|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}_{k-1}) - \texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1})\|_2 \\
&\leq \|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1}) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1})\|_2 + c_\phi \|\boldsymbol{y}_{k-1} - \boldsymbol{y}'_{k-1}\|_2
\end{aligned}
$$

If there exists a constant K, independent of k, ϕ, ϕ', such that

$$
\|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1}) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}'_{k-1})\|_2 \leq K \|\phi - \phi'\|_2, \tag{62}
$$

then we can obtain from $\boldsymbol{y}_0 = \boldsymbol{y}'_0$ that

$$
\begin{aligned}
\|\boldsymbol{y}_k - \boldsymbol{y}'_k\|_2 &\leq v \|\phi - \phi'\|_2 + c_\phi \|\boldsymbol{y}_{k-1} - \boldsymbol{y}'_{k-1}\|_2 \\
&\leq K \|\phi - \phi'\|_2 \sum_{i=1}^k c_\phi^{i-1} = \frac{1 - c_\phi^k}{1 - c_\phi} K \|\phi - \phi'\|_2.
\end{aligned}
$$

The fact that $c_\phi \leq c_0 < 1$ for some constant c_0 implies that the magnitude of sensitivity is $\mathcal{O}(1 - (\inf_{\phi \in \Phi} c_\phi)^k)$.

Now it remains to establish the estimate Eq. 62. For each $k \in \mathbb{N}$, $\phi = \{W_1^1, W_1^1, W^2, \ldots, W^L, V\}$ and $l = 2, \cdots, L$, we introduce the notation

$$
f_\phi^l := W^l \sigma\left(W^{l-1} \cdots W^2 \sigma\left(W_1^1 \boldsymbol{y}_k + W_2^1 \boldsymbol{g}_k\right)\right), \tag{63}
$$

with $f^1_\phi = W_1^1 \boldsymbol{y}_k + W_2^1 \boldsymbol{g}_k$. Then we have for each $l = 1, \cdots, L$ that

$$\|f^l_\phi\|_2 \le \prod_{j=2}^{l} \|W^j\|_2 \left(\|W_1^1 + W_2^1 Q\|_2 \|\boldsymbol{y}_k\|_2 + \|W_2^1\|_2 \|\boldsymbol{b}\|_2\right) = c_l \|\boldsymbol{y}_k\|_2 + \hat{c}_l \|\boldsymbol{b}\|_2, \tag{64}$$

with the constants $c_l := \big(\prod_{j=2}^{l} \|W^j\|_2\big)\|W_1^1 + W_2^1 Q\|_2$, $\hat{c}_l := \big(\prod_{j=2}^{l} \|W^j\|_2\big)\|W_2^1\|_2$ for all $l = 1, \ldots, L$. Then by induction, we can see that

$$\begin{aligned}
\|f^L_\phi - f^L_{\phi'}\|_2 &= \|W^L \sigma(f^{L-1}_\phi) - W'^L \sigma(f^{L-1}_{\phi'})\|_2 \\
&\le \|W^L - W'^L\|_2 \|f^{L-1}_{\phi'}\|_2 + \|W^L\|_2 \|f^{L-1}_\phi - f^{L-1}_{\phi'}\|_2 \\
&\le \|W^L - W'^L\|_2 \|f^{L-1}_{\phi'}\|_2 + \|W^L\|_2 \Bigg(\|W^{L-1} - W'^{L-1}\|_2 \|f^{L-2}_{\phi'}\|_2 \\
&\quad + \|W^{L-1}\|_2 \|f^{L-2}_\phi - f^{L-2}_{\phi'}\|_2 \Bigg) \\
&\le \sum_{l=2}^{L} \left(\prod_{j=l+1}^{L} \|W^j\|_2 \right) \|W^l - W'^l\|_2 \|f^{l-1}_{\phi'}\|_2 + \left(\prod_{l=2}^{L} \|W^l\|_2 \right) \|f^1_\phi - f^1_{\phi'}\|_2.
\end{aligned}$$

Thus we have that

$$\begin{aligned}
&\|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}_k) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}_k)\|_2 = \|V\sigma(f^L_\phi) - V'\sigma(f^L_{\phi'})\|_2 \\
&\le \|V - V'\|_2 \|f^L_{\phi'}\|_2 + \|V\|_2 \|f^L_\phi - f^L_{\phi'}\|_2 \\
&\le \|V - V'\|_2 \|f^L_{\phi'}\|_2 + \|V\|_2 \Bigg[\sum_{l=2}^{L} \left(\prod_{j=l+1}^{L} \|W^j\|_2 \right) \|W^l - W'^l\|_2 \|f^{l-1}_{\phi'}\|_2 \\
&\quad + \left(\prod_{l=2}^{L} \|W^l\|_2 \right) \|f^1_\phi - f^1_{\phi'}\|_2 \Bigg].
\end{aligned}$$

Furthermore, we see that

$$\begin{aligned}
\|f^1_\phi - f^1_{\phi'}\|_2 &= \|(W_1^1 + W_2^1 Q)\boldsymbol{y}_k + W_2^1 \boldsymbol{b} - (W_1'^1 + W_2'^1 Q)\boldsymbol{y}_k + W_2'^1 \boldsymbol{b}\|_2 \\
&\le \|W_1^1 - W_1'^1 + (W_2^1 - W_2'^1)Q\|_2 \|\boldsymbol{y}_k\|_2 + \|W_2^1 - W_2'^1\|_2 \|\boldsymbol{b}\|_2 \\
&\le \|W_1^1 - W_1'^1\| \|\boldsymbol{y}_k\|_2 + \|W_2^1 - W_2'^1\| (\|Q\|_2 \|\boldsymbol{y}_k\|_2 + \|\boldsymbol{b}\|_2),
\end{aligned}$$

from which we can conclude that

$$\begin{aligned}
&\|\texttt{RNNcell}_\phi(Q, \boldsymbol{b}, \boldsymbol{y}_k) - \texttt{RNNcell}_{\phi'}(Q, \boldsymbol{b}, \boldsymbol{y}_k)\|_2 \\
&\le \|f^L_{\phi'}\|_2 \|V - V'\|_2 + \sum_{l=2}^{L} \left[\|V\|_2 \left(\prod_{j=l+1}^{L} \|W^j\|_2 \right) \|f^{l-1}_{\phi'}\|_2 \right] \|W^l - W'^l\|_2 \\
&\quad + \|V\|_2 \left(\prod_{l=2}^{L} \|W^l\|_2 \right) \Big[\|W_1^1 - W_1'^1\| \|\boldsymbol{y}_k\|_2 + \|W_2^1 - W_2'^1\| (\|Q\|_2 \|\boldsymbol{y}_k\|_2 + \|\boldsymbol{b}\|_2) \Big].
\end{aligned}$$

Note that we have assumed that the set of parameters Φ is a compact subset of the stable region and $(Q, \boldsymbol{b}) \in \mathcal{S}^{d\times d}_{\mu,L} \times \mathcal{B}$ are bounded, which imply that for all $\phi, \phi' \in \Phi$, the corresponding outputs $(\boldsymbol{y}_k)_{k\in\mathbb{N}}$ and $(\boldsymbol{y}'_k)_{k\in\mathbb{N}}$ are uniformly bound, and hence $\|f^l_{\phi'}\|_2$ is bounded for all k and $l = 1, \ldots, L$ (see Eq. 64). Consequently, we see there exists a constant K such that Eq. 62 is satisfied. This finishes the proof of the desired sensitivity result. □

(IV) Convergence. For the convergence of $\texttt{RNN}^k_\phi$, we can only give the best case guarantee. It is easy to see that with the following choice of ϕ, $\texttt{RNN}^k_\phi$ can represent $\texttt{GD}^k_s$:

$$V = [I, -I], \quad W_1^1 = [I; -I]^\top, \quad W_1^2 = [-sI; sI]^\top, \quad W^l = I \text{ for } l = 2, \cdots, L. \tag{65}$$

Therefore, for the best case, $\texttt{RNN}^k_\phi$ can converge at least as fast as $\texttt{GD}^k_s$.

H. Experiment Details

Here we state the configuration details of the experiments.

- Convexity and smoothness. They are set to be $\mu = 0.1$ and $L = 1$, respectively.
- Dataset. 10000 pairs of $(\boldsymbol{x}, \boldsymbol{b})$ are generated in the following way: 10000 many $\boldsymbol{x}$ are uniformly sampled from $[-5,5]^{10} \times \mathcal{U}^{5\times 5}$, where $\mathcal{U}^{5\times 5}$ denotes the space of all 5×5 unitary matrices. Each input $\boldsymbol{x}$ actually is a tuple $\boldsymbol{x} = (\boldsymbol{z}_{\boldsymbol{x}}, U_{\boldsymbol{x}})$ where $\boldsymbol{z}_{\boldsymbol{x}} \in [-5,5]^{10}$ and $U_{\boldsymbol{x}}$ is unitary. 10000 many $\boldsymbol{b}$ are uniformly sampled from $[-5,5]^5$. These 10000 pairs are viewed as the whole dataset.
- Training set S_n. During training, n samples are randomly drawn from these 10000 data points as the training set. The labels of these training samples are given by $\boldsymbol{y} = \texttt{Opt}(Q^*(\boldsymbol{x}), \boldsymbol{b})$.
- More details on $Q^*(\boldsymbol{x})$. As mentioned before, each $\boldsymbol{x}$ is a tuple $\boldsymbol{x} = (\boldsymbol{z}_{\boldsymbol{x}}, U_{\boldsymbol{x}})$. Then we implement $Q^*(\boldsymbol{x}) = U_{\boldsymbol{x}} \text{diag}([g^*(\boldsymbol{z}_{\boldsymbol{x}}), \mu, L]) U_{\boldsymbol{x}}^\top$, where g^* is a 2-layer dense neural network with hidden dimension 3, output dimension 3, and with randomly fixed parameters. Note that in the final layer of g^*, there is a sigmoid-activation that scales the output to the range $[0,1]$ and then the range is further re-scaled to $[\mu, L]$. Finally, $g^*(\boldsymbol{z}_{\boldsymbol{x}})$ is concatenated with $[\mu, L]$ to form a 5-dimensional vector with smallest and largest value to be μ and L respectively. This vector represents the eigenvalues of $Q^*(\boldsymbol{x})$.
- Architecture of Q_θ. Q_θ has the same form as $Q^*(\boldsymbol{x})$, except that the network g^* in Q^* becomes g_θ in Q_θ. That is, $Q_\theta(\boldsymbol{x}) = U_{\boldsymbol{x}} \text{diag}([g_\theta(\boldsymbol{z}_{\boldsymbol{x}}), \mu, L]) U_{\boldsymbol{x}}^\top$. Here g_θ is also a 2-layer dense neural network with output dimension 3, but the hidden dimension can vary. In the reported results, when we say hidden dimension=0, it means g_θ is a one-layer network.

For the experiments that compare $\texttt{RNN}_\phi^k$ with $\texttt{GD}_\phi^k$ and $\texttt{NAG}_\phi^k$, they are conducted under the 'learning to learn' scenario, with the following modifications compared to the above setting.

- Dataset. Instead of sampling $(\boldsymbol{x}, \boldsymbol{b})$, here we directly sample the problem pairs $(Q, \boldsymbol{b})$. Similarly, 10000 pairs of $(Q, \boldsymbol{b})$ are sampled uniformly from $\mathcal{S}_{\mu,L}^{10\times 10} \times [-5,5]^{10}$.
- Architecture of $\texttt{RNN}_\phi^k$. For each cell in $\texttt{RNN}_\phi^k$, it is a 4-layer dense neural network with hidden dimension 20-20-20.

For all experiments, each model has been trained by both ADAM and SGD with learning rate searched over [1e-2,5e-3,1e-3,5e-4,1e-4], and only the best result is reported. Furthermore, error bars are produced by 20 independent instantiations of the experiments. The experiments are mainly run parallelly (since we need to search the best learning rate) on clusters which have 416 nodes where on each node there are 24 Xeon 6226 CPU @ 2.70GHz with 192 GB RAM and 1x512 GB SSD.