ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889376

Epigraph Based Multilevel Optimization (EMO) for Enhancing Chain-of-Thought Reasoning Capabilities

Songtao Lu[⋆], *Yanna Ding*[†,‡], *Lior Horesh*[†], *Jianxi Gao*[‡], *Malik Magdon-Ismail*[‡]

[⋆]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
[†]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, New York 10598, USA
[‡]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, USA

ABSTRACT

Chain-of-thought (CoT) reasoning applies to complex tasks with multiple intermediate steps, a key feature of large language models. Recent studies have revealed CoT as a composition of in-context filtering and learning. This paper proposes a unified framework for CoT optimization that exploits the nested problem structure to formulate training as multilevel optimization. Each intermediate reasoning step is a distinct optimization level. We develop an epigraph-based multilevel optimization (EMO) method to iteratively find the optimal solution for this class of problems. Experiments using GPT-2 show that the proposed EMO achieves the lowest generalization errors across all intermediate steps compared to state-of-the-art, highlighting the importance of nested optimization approaches for CoT reasoning.

***Index Terms*—** Chain-of-thought (CoT), multilevel optimization (MLO), epigraph-based multilevel optimization (EMO), large language models (LLMs)

1. INTRODUCTION

Large language models (LLMs) with transformer architectures [1, 2] demonstrate complex reasoning that captures dependencies over long sequences [3]. LLMs excel at in-context learning, meaning that once pre-trained, they can implicitly fine-tune based on new prompts and generate high-quality sentences without updating the model parameters [4, 5]. Self-attention is the mechanism that prioritizes relevant features from the prompt, aligning them with learned features to produce predictions [6–8].

LLM reasoning is enhanced when step-by-step instructions are given [9]. For example, to learn the relationship $y = (2x)^2$, a composition of $2x$ and x^2, a prompt like $p : (x = 1, y = 4)$ leaves the mapping from x to y ambiguous. If intermediate steps are given, $p : (x = 1, s = 2x = 2, y = s^2 = 4)$, the relationship is easier to learn due to the reduced ambiguity. The richer prompt makes the model predictions more accurate and informative with clear in-context meaning. A reasoner's ability to reveal intermediate steps is referred to as *chain-of-thought* (CoT) reasoning [9]. Circuit complexity results [10] show that CoT prompting is necessary for complex problems, such as dynamic programming (bounded-depth Transformers fail unless the model size grows super-polynomially in input length). By increasing the number of intermediate steps Transformers become more expressive [11], e.g., Boolean circuits of size K can be represented using $\mathcal{O}(K)$ CoT steps [12].

Even though the reasoning capability can be enhanced by the CoT-style recursive decoding process, Transformer-based models are still trained using traditional methods. As discussed in the existing literature [13], CoT model training primarily focuses on minimizing the sum of inference errors at each step, neglecting the chain structure of the reasoning process. Motivated by the decoding procedure, it is more reasonable to formulate CoT as a multilevel programming problem, as the quality of each step depends on all preceding ones. Therefore, the accuracy of the first step is more critical than that of subsequent steps, with the same logic applying to the second, third, and so on. However, solving multilevel optimization (MLO) problems is challenging. Due to the nested structure, computing the derivatives at each level requires higher-order information from all lower-level loss functions [14], such as Jacobian or Hessian inverse matrices, even in the bilevel case [15]. Several approaches, including approximate implicit differentiation [16], recursive matrix inversion-based gradient descent [17], the penalty-based method [18], and the primal-dual method [19], have been developed to solve bilevel optimization problems. However, when applying these methods to MLO, it becomes necessary to compute complex implicit gradients using second-order information from the loss functions [20, 21], which makes the algorithms impractical for large-scale problems due to increased computational complexity. Additionally, the nested structure in MLO may cause the conditions required by existing methods to fail, e.g., penalty methods. To the best of our knowledge, none of the existing methods rely solely on first-order information to solve MLO problems.

In this paper, we propose a nested structured training framework to improve CoT reasoning capabilities. By leveraging constrained optimization techniques, we introduce an epigraph-based multilevel optimization (EMO) algorithm to solve this problem, applying subgradient descent only to the maximum loss function across all levels. Theoretical discussions show that EMO effectively addresses nested multilevel model training problems, where the loss functions at all levels are nonconvex and smooth, while providing theoretical guarantees of convergence. Numerical experiments with GPT-2 demonstrate that when trained using this nested structure with EMO, the model accurately outputs all intermediate inference steps for both in-distribution and out-of-distribution (OOD) prompts, outperforming the traditional single-level empirical risk minimization strategy. The major contributions of this work are highlighted as follows:

- A new nested framework for CoT model training with immediate reasoning steps is established based on an MLO problem.
- Our proposed epigraph subgradient algorithm is the first method that finds an ϵ-approximate solution to the reformulated MLO problem with K levels in $\mathcal{O}(\log^K(1/\epsilon)/\epsilon^4)$ iterations.
- Experiments show EMO achieves the best efficiency and test errors compared to the state-of-the-art.

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on April 18,2025 at 22:00:56 UTC from IEEE Xplore. Restrictions apply.

2. PROBLEM FORMULATION OF COT

We use the CoT setup in [13]. The ground truth target function is a reasoning chain that maps from features $\{x_i\}$ to labels $\{y_i\}$, denoted as $\mathcal{F}$, where $\mathcal{F} = \mathcal{F}_K \circ \cdots \circ \mathcal{F}_2 \circ \mathcal{F}_1$. In a standard learning setup, the prompt containing the n training data points plus the test query is given by $p = \{(x_1, y_1), \ldots, (x_n, y_n), (x_q, ?)\}$, where $y_i = \mathcal{F}(x_i)$. In the CoT setup, the training data also contains the labels of the intermediate functions and the in-context prompt becomes

$$p = \{(x_1, \mathbf{s}_1), \ldots, (x_n, \mathbf{s}_n), (x_q, ?)\}, \tag{1}$$

where the vector $\mathbf{s}_i = [s_i^1, \ldots, s_i^K]$ contains the labels at every level of the ground truth CoT target $\mathcal{F}$ for input x_i. Let $s_i^0 = x_i$. Then,

$$s_i^k = \mathcal{F}_i(s_i^{k-1}) \quad \text{for} \quad k = 1, \ldots, K. \tag{2}$$

Here, that s_i^k represents the kth intermediate step in the CoT, $y_i = s_i^K$, and the function $\mathcal{F}_k$ at the kth step is in some function class $\mathbb{F}_k$.

We approximate $\mathcal{F}$ by a Transformer $\texttt{TF}_\theta$ having parameters θ. The Transformer predicts on input x_i to get $\hat{s}_i^1$, then on input $(x_i, \hat{s}_i^1)$ to get $\hat{s}_i^2$ and so on until it gets $\hat{s}_i^K$. That is, $\hat{\mathbf{s}}_i(\theta) = [\hat{s}_i^1, \ldots, \hat{s}_i^K]$ with $\hat{s}_i^1(\theta) = \texttt{TF}_\theta(x_i)$ and

$$\hat{s}_i^k(\theta) = \texttt{TF}_\theta(x_i, \hat{s}_i^1, \ldots, \hat{s}_i^{k-1}) \quad \text{for} \quad k = 2, \ldots, K. \tag{3}$$

To recover θ, we minimize a loss. Unlike in [13], we define a loss that exploits the multilevel structure of the CoT process. Intuitively, we want to match every intermediate step of the ground truth, that is to minimize $\max\{\|\hat{s}_i^1 - s_i^1\|, \ldots, \|\hat{s}_i^K - s_i^K\|\}$ summed over training data. To do so we formulate a MLO problem. Let f_k define the level k objective,

$$f_k(\theta) = \mathbb{E}_{\{x_i\}_{i=1}^n, \{\mathcal{F}_j\}_{j=1}^k} \left[\frac{1}{n} \sum_{i=1}^{n} \frac{1}{k} \sum_{j=1}^{k} \ell(\hat{s}_i^j(\theta), s_i^j) \right], \tag{4}$$

where $\ell(\cdot, \cdot)$ is a pairwise loss function such as the absolute or cross-entropy error. The objective $f_k(\theta)$ captures the average error being made in the first k intermediate steps. Let θ_1 minimize $f_1(\theta)$, attaining minimum objective $\mathcal{E}_1 \triangleq \min_\theta f_1(\theta)$. The model θ_1 has maximum accuracy for the first intermediate step in the reasoning chain. For $k > 1$ we define $\theta_k^\star$ and $\mathcal{E}_k$ recursively,

$$\mathcal{E}_k = \min_\theta f_k(\theta) \quad \text{s.t. } f_j(\theta) \leq \mathcal{E}_j \text{ for } j = 1, \ldots, k-1 \tag{5}$$

and $\theta_k^\star$ are the parameters that attain the minimum in the constrained optimization above. As seen in (5), the model learns the intermediate steps level by level. Indeed, the model with parameters $\theta_k^\star$ approximates intermediate steps $1, \ldots, k$. The full learned model is $\theta_\star = \theta_K^\star$. Note that learning the level-k model $\theta_k^\star$ while requiring that all earlier levels are optimal is a stringent constraint, especially if the model $\texttt{TF}_\theta$ lacks sufficient expressiveness. By adding some slack to this constraint, in the next section we reformulate a relaxed version of the problem allowing us to develop an efficient method to find an approximate solution using only gradients.

3. NESTED MULTILEVEL OPTIMIZATION

Directly solving (5) is nontrivial even for the bilevel case. Instead, we can relax this inequality to $f_j(\theta) - \mathcal{E}_j \leq \varepsilon$, where $\varepsilon > 0$ is the relaxation variable, such that there exists an interior point satisfying this constraint, i.e., the Slater condition always holds [22].

By recursively applying this property *backward* from the Kth to the 1st level, the original MLO problem (5) can be reformulated as

$$\min_\theta \quad f_K(\theta) \tag{6a}$$

$$\text{s.t.} \quad g_{K-1}(\theta) \leq 0, \text{ ... s.t. } g_k(\theta) \leq 0, \ldots \text{ s.t. } g_1(\theta) \leq 0 \tag{6b}$$

where inequality constraint functions

$$g_k(\theta) \triangleq f_k(\theta) - \mathcal{E}_k - \varepsilon \leq 0, \quad \forall k. \tag{7}$$

Note that unless the optimal solution sets of these multilevel problems overlap, classic single-level constrained problems are fundamentally different from multilevel ones. We impose an order in the optimization process, where the kth-level optimization depends on the $(k-1)$th-level.

Major Challenge. One of the most straightforward approaches might be the penalty method, which simply penalizes the constraints in the objective, transforming the problem into a single-level optimization problem. This method works well for constrained optimization or even bilevel optimization, but it may face challenges in selecting an appropriate step size if it is adopted for solving MLO. The reason is that when there are multiple levels of constraints, for example, three levels, the penalized objective can involve higher-order terms of the penalty parameters. In this case, the objective might take the form $f_3(\theta) + \rho g_2(\theta) + \rho^2 g_1(\theta)$, which affects the gradient Lipschitz constant of the loss function. This, in turn, results in the step-size requirement for ensuring convergence of gradient descent-type algorithms being on the order of $\mathcal{O}(1/\rho^k)$. It is evident that when $\rho > 1$, the required step size will be very small, which hinders the learning process.

New Algorithm. We propose an epigraph reformulation [22] of this problem by introducing a slack variable as follows.

$$\min_{\theta, z} \quad z \tag{8a}$$

$$\text{s.t.} \quad f_K(\theta) \leq z, g_{K-1}(\theta) \leq 0, \text{ s.t.} \ldots, \text{ s.t. } g_1(\theta) \leq 0. \tag{8b}$$

Rather than minimize the original loss function, we minimize an upper bound. Using the epigraph form, we rewrite $\min_{\theta,z} z$, s.t. $g(\theta, z) \leq 0$ as $\min_z z$, s.t. $\min_\theta g(\theta, z) \leq 0$ [23, Theorem 3], which corresponds to the original form of the problem as

$$\min_{z \geq 0} \quad z \tag{9a}$$

$$\text{s.t. } \min_\theta \widetilde{g}_{K-1}(\theta) \triangleq \max\{g_{K-1}(\theta), f_K(\theta) - z\} \leq 0, \tag{9b}$$

$$\text{s.t.} \quad g_{K-2}(\theta) \leq 0, \ldots \text{ s.t.} \quad g_1(\theta) \leq 0. \tag{9c}$$

Recursively applying this rule *forward* from level 1 up to K gives

$$\min_{z_1 \geq 0} \quad z_1 \tag{10a}$$

$$\text{s.t. } \min_{z_2 \geq 0} z_2, \text{ s.t. } \ldots, \tag{10b}$$

$$\text{s.t. } \min_\theta \widetilde{g}(\theta) \triangleq \max_k \Bigg\{ g_1(\theta), \ldots, g_k(\theta) - \sum_{i=k}^{K-1} z_i, \ldots, f_K(\theta) - \sum_{i=1}^{K-1} z_i \Bigg\} \leq 0. \tag{10c}$$

Update of θ. The last level optimization problem finds a θ such that the maximum function value among $g_1(\theta), \ldots, g_k(\theta) - \sum_{i=k}^{K-1} z_i$, and $f_K(\theta) - \sum_{i=1}^{K-1} z_i$ is minimized. Let $k^\star(\theta) = \arg\max_k \{g_1(\theta), \ldots, g_k(\theta) - \sum_{i=k}^{K-1} z_i, \ldots, f_K(\theta) - \sum_{i=1}^{K-1} z_i\}$ be the index of the loss function that has the maximum value. One of the most computationally efficient algorithms for optimizing the maximum of a set of functions is the (stochastic) subgradient descent method. Let $\partial f(\cdot)$ denote the subdifferential of $f(\cdot)$. Since $\mathcal{E}_k$ is not a function of θ, we can update the variables by applying the $k^\star(\theta^{(r)})$th-level loss function as follows:

$$\theta^{(r+1)} = \theta^{(r)} - \alpha \zeta_g^{(r)} = \theta^{(r)} - \alpha \zeta_f^{(r)} \tag{11}$$

where r indexes the iterations, α is the learning rate, $\zeta_g^{(r)} \in \partial_\theta \widetilde{g}(\theta^{(r)})$, $\zeta_f^{(r)} \in \partial_\theta \widetilde{f}(\theta^{(r)})$, and $\widetilde{f}(\cdot) \triangleq \max_k \{f_1(\cdot), \ldots, f_K(\cdot)\}$.

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on April 18,2025 at 22:00:56 UTC from IEEE Xplore. Restrictions apply.

Update of $\{z_k, \forall k\}$. Once the above optimization process is complete (i.e., either when the maximum value of all loss functions is less than 0 for $(z_1, \ldots, z_{K-1})$, or the maximum number of iterations is reached), we can further adjust the values of $\{z_k, \forall k\}$ such that θ can be optimized for all levels of the loss functions. One issue here is that $\mathcal{E}_k$ is generally unknown; however, a bisection algorithm can be applied to find z_k for all k and ensure convergence. Specifically, we initialize $z_1^{(0)}, \ldots, z_{K-1}^{(0)}$ with large positive values. After updating θ (for several epochs), we check if condition (10c) is satisfied (or if the maximum number of iterations is reached) and bisect the interval $[0, z_{K-1}^{(0)}]$. We alternate between updating θ and bisecting z_{K-1} until the stopping criterion is met. Then, we move to the next level and repeat the process. The complete algorithm is summarized in Algorithm 1.

Algorithm 1: Pseudo-Code of EMO Algorithm

Data: prompts $(x_i, s_i^1, \ldots, s_i^{K-1}, y_i)_{i=1}^n$
Result: learned model parameters $\theta^{(T)}$
Initialize $\theta^{(0)}$, $\{z_k^{(0)}, \forall k\}$
for *Choose* z_1 *by the bisection algorithm* **do**
  for ... **do**
    for *Choose* z_{K-1} *by the bisection algorithm* **do**
      while *condition* (10c) *is not satisfied or max iterations reached* **do**
        Select $k^\star(\theta^{(r)})$ based on $\widetilde{g}(\theta^{(r)})$
        Update $\theta^{(r+1)}$ by (11)
      end
    end
  end
end

Iteration Complexity Analysis of EMO. The complexity of EMO is of the same order as that of the standard subgradient optimization algorithm, up to a logarithmic factor, making it well-suited for solving large-scale problems. Specifically, we present the following theoretical results.

Theorem 1. *Assume that the loss functions are smooth and convex. Suppose that the sequence* $\{\theta^{(r)}\}_{r=0}^T$ *is generated by the EMO algorithm and the learning rate is* $\alpha \sim \mathcal{O}(1/\sqrt{T})$*. Then, EMO achieves a nearly* ϵ*-stationary point of problem* (10) *if the total number of iterations* T *satisfies* $T \geq \mathcal{O}(\log^K(1/\delta)/\epsilon^4)$*, where* δ *is the tolerance error required in the stopping criterion of the bisection oracle.*

Proof (Sketch) Bisection achieves error δ in $\mathcal{O}(\log(1/\delta))$ iterations in each level, hence $K-1$ levels requires $\mathcal{O}(\log^K(1/\delta))$ iterations. For smooth loss functions with learning rate $\alpha \sim \mathcal{O}(1/\sqrt{T})$, subgradient algorithms achieve ϵ-stationary points in $\mathcal{O}(1/\epsilon^4)$ iterations [24]. Multiplying these two complexities gives Theorem 1.

Reduction of Computational Complexity. One simple way to reduce the complexity is to set all z_k to a single variable z, so that problem (10) reduces to the following bilevel problem.

$$\min_{z \geq 0} \quad z \tag{12a}$$

$$\text{s.t.} \ \min_{\theta} \max_{k} \big\{ g_1(\theta), \ldots, g_k(\theta) - (K-k)z, \ldots, f_K(\theta) - (K-1)z \big\} \leq 0, \tag{12b}$$

which directly decreases the complexity from $\mathcal{O}(\log^K(1/\epsilon)/\epsilon^4)$ to $\mathcal{O}(\log(1/\epsilon)/\epsilon^4)$ with $\delta = \epsilon$. Note that the term $(K-k)z$ still maintains the nested structure of this problem, emphasizing the priority of enforcing the constraint at the $(k-1)$th level compared to the kth level.

Variants of EMO. The proposed EMO can also be adapted in other ways. For example, using a Moreau envelope-based reformulation of the loss functions at each level results in a variant of EMO as follows.

EMO-M. Some recent works on bilevel optimization introduce a Moreau envelope-based reformulation [25, 26] for the lower-level loss function, which provides an alternative approach to estimating the minimum value $\mathcal{E}_k$ while preserving the nested structure of the original problem. Specifically, we can revise the kth level problem as $f_k(\theta) - \mathcal{E}_k^\gamma(\theta) \leq \varepsilon$, where $\mathcal{E}_k^\gamma(\theta) \triangleq \min_\phi f_k(\phi) + (2\gamma)^{-1}\|\phi - \theta\|^2$. When γ is small, this loss function becomes strongly convex with respect to ϕ, allowing us to easily obtain $v_k^\gamma(\theta)$ by applying several steps of gradient descent on ϕ. The advantage is that it eliminates the need for a bisection search for z, while the downside is the additional computational burden of the inner loop oracle required to obtain $\mathcal{E}_k^\gamma(\theta)$, which is only estimated approximately.

EMO-G. Instead of evaluating function values in condition (10c), we can use the size of the gradient as a metric to determine which block should be updated, i.e., $k^\star(\theta) = \arg\max_k \{\|\nabla_\theta f_1(\theta)\|, \ldots, \|\nabla_\theta f_K(\theta)\|\}$ [24]. This approach eliminates the need to search for z, but it loses the nested structure of the original problem, reducing it to a switching gradient method for solving nonlinear constrained optimization problems. We term this algorithm EMO-G, to serve as a baseline.

4. EXPERIMENTS

We empirically evaluate the performance of applying EMO and its variants for CoT inference problems compared to the vanilla single-level training strategy (also known as CoT-I [13]), CoT-I/O with the sum of all step-wise loss functions [13], and the penalty-based method. During inference, given the prompt x_q (from either the in-distribution or OOD dataset) as input, the learned model, optimized with parameter $\theta_\star$, recursively outputs all intermediate steps as follows: $\texttt{TF}_{\theta_\star}(x_q, \hat{s}^1, \ldots, \hat{s}^{k-1}) = \hat{s}^k$ for all k.

Datasets. We generate chains using 3-layer MLPs with input $x_i \in \mathbb{R}^d$ where $d = 10$, hidden features $s_i^1, s_i^2 \in \mathbb{R}^q$ with $q = 4$, and output $y_i \in \mathbb{R}$. Here, $s_i^1 = \mathcal{F}_1(x_i) \triangleq \text{ReLU}(W_1 x_i)$, $s_i^2 = \mathcal{F}_2(s_i^1) \triangleq \text{ReLU}(W_2 s_i^1)$, and $y_i = \mathcal{F}_3(s_i^2) \triangleq w_3^\top s_i^2$ for some weight matrices $W_1 \in \mathbb{R}^{q \times d}$, $W_2 \in \mathbb{R}^{q \times q}$, and $w_3 \in \mathbb{R}^q$. The reasoning chain is the composition of the three functions, i.e., $y_i = \mathcal{F}(x_i) = w_3^\top \text{ReLU}(W_2 \text{ReLU}(W_1 x_i))$. The distribution of the feature vectors and task parameters is zero-mean Gaussian: $x_i \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d)$; each entry of W_i is randomly sampled from $\mathcal{N}(0, 2/q)$ and $w_3 \sim \mathcal{N}(0, I_q)$ for each prompt.

In addition to the in-distribution test dataset, we create OOD data samples to evaluate the robustness of the optimization algorithms against distribution shifts. For the training and in-distribution test datasets, the covariance matrix is fixed to the identity matrix. For the OOD dataset, we draw the in-context input vectors from $x_\text{q} \sim \mathcal{N}(0, c\Lambda)$, where Λ is diagonal with $\lambda_i \overset{i.i.d.}{\sim}$ Exponential(1) and $c = 4$, following [7].

Training. All the experiments use the GPT-2 model [27] with three transformer blocks. We adopt Adam [28] to optimize the model over 50k epochs with batch size 64. For each prompt, n feature vectors are randomly sampled and then labeled by a set of randomly generated task parameters. During training, n increases from 1 to 100 to cover varying prompt lengths. We report the square loss as our metric for step-wise error, i.e., $\ell(\hat{s}_i^k, s_i^k) = \|\hat{s}_i^k - s_i^k\|_2^2$.

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on April 18,2025 at 22:00:56 UTC from IEEE Xplore. Restrictions apply.

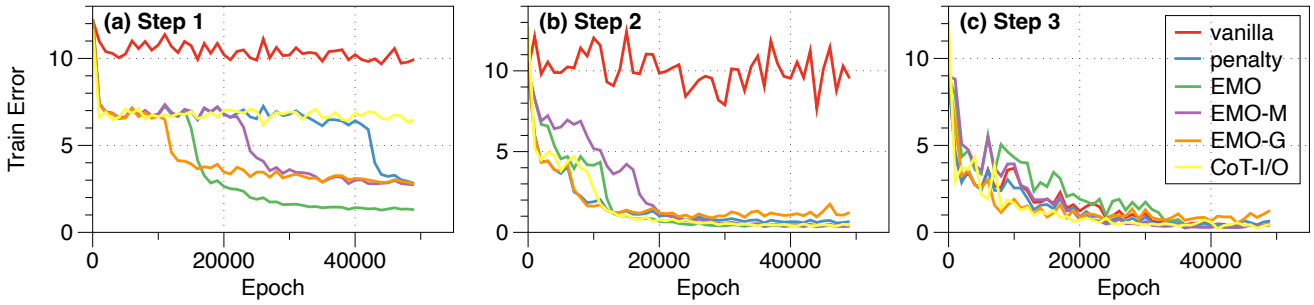

Fig. 1: Performance comparison of EMO, EMO-M, EMO-G, vanilla training, CoT-I/O, penalty method in terms of training errors.

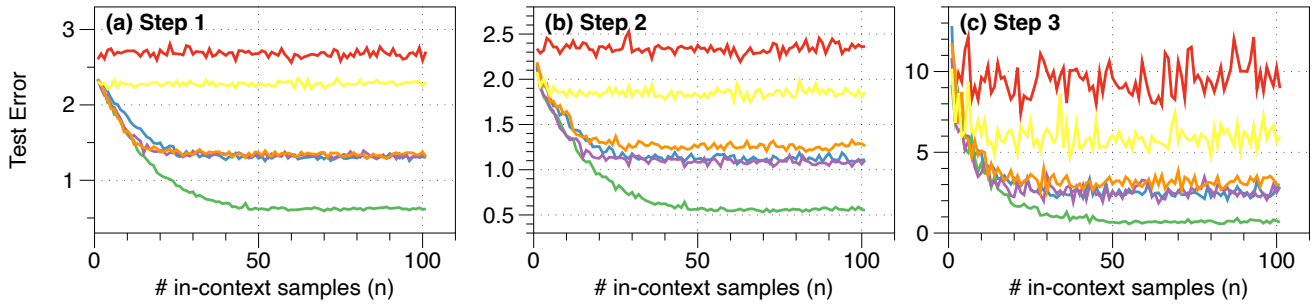

Fig. 2: Performance comparison of EMO, EMO-M, EMO-G, vanilla training, CoT-I/O, penalty method in terms of in-distribution test errors.

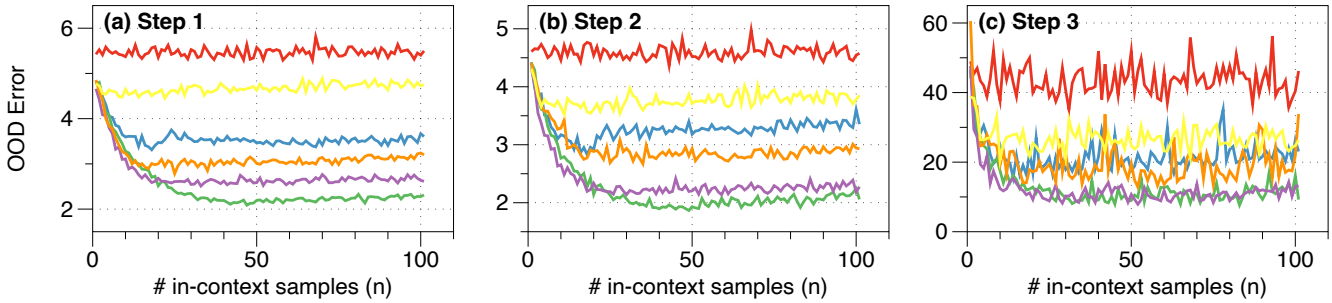

Fig. 3: Performance comparison of EMO, EMO-M, EMO-G, vanilla training, CoT-I/O, penalty method in terms of OOD test errors.

Numerical Results. The performance for training, testing, and OOD scenarios is demonstrated in Figure 1, Figure 2, Figure 3, respectively. As shown in Figure 1, EMO achieves the lowest training error at all levels and maintains a competitive convergence rate compared to other baseline methods. EMO-M, while showing relatively worse performance in terms of achievable training errors compared to the other methods, still outperforms the penalty method, particularly for the first reasoning step.

Due to the page limit, we omit the test error vs. epochs and instead include the test error vs. prompt length n, which better reflects the generalization performance of the nested machine models, consistent with the experiment in [13]. It is observed that as the prompt length increases, CoT performs better, particularly EMO, achieving the lowest test error. This improvement is attributed to the nested training structure, which emphasizes minimizing the loss function level by level, or equivalently, step by step. After a certain n, the test errors for all methods stabilize as n increases, which is consistent with existing theoretical generalization analyses [13, 29].

Interestingly, the OOD errors obtained by all methods, except the vanilla one, exhibit a slight U-curve pattern, indicating that inference error accumulates beyond a certain prompt length due to the mismatch between the training and test prompts. It can also be observed that in this scenario, both EMO and EMO-M show significant improvement over the other baselines, underscoring the importance of step-wise model parameter optimization over the vanilla summation formulation.

5. CONCLUDING REMARKS

We proposed a unified nested MLO framework for CoT inference. To the best of our knowledge, this is the first mathematical formulation established for designing CoT-structured machine learning models, where each level of the optimization process enhances the quality of the subsequent step prediction, enabling step-by-step inference. We further introduce an epigraph-based MLO reformulation for this class of problems to design a first-order method that can find stationary points of this problem with provable convergence guarantees. Our numerical results on the GPT-2 language model demonstrate that the proposed nested training strategy maintains high CoT inference quality at each step and outperforms all baseline models in terms of overall training speed and generalization performance.

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on April 18,2025 at 22:00:56 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] A. Vaswani, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[3] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al., "Sparks of artificial general intelligence: Early experiments with GPT-4," *arXiv preprint arXiv:2303.12712*, 2023.

[4] A. Singh, S. Chan, T. Moskovitz, E. Grant, A. Saxe, and F. Hill, "The transient nature of emergent in-context learning in transformers," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[5] J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov, "Transformers learn in-context by gradient descent," in *Proceedings of International Conference on Machine Learning (ICML)*, 2023, pp. 35151–35174.

[6] Y. Tian, Y. Wang, B. Chen, and S. S. Du, "Scan and snap: Understanding training dynamics and token composition in 1-layer transformer," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023, pp. 71911–71947.

[7] R. Zhang, S. Frei, and P. L. Bartlett, "Trained transformers learn linear models in-context," *Journal of Machine Learning Research*, vol. 25, no. 49, 2024.

[8] H. Li, M. Wang, S. Lu, X. Cui, and P.-Y. Chen, "How do nonlinear transformers learn and generalize in in-context learning?," in *Proceedings of International Conference on Machine Learning (ICML)*, 2024.

[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022, vol. 35, pp. 24824–24837.

[10] G. Feng, B. Zhang, Y. Gu, H. Ye, D. He, and L. Wang, "Towards revealing the mystery behind chain of thought: a theoretical perspective," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[11] W. Merrill and A. Sabharwal, "The expresssive power of transformers with chain of thought," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2024.

[12] Z. Li, H. Liu, D. Zhou, and T. Ma, "Chain of thought empowers transformers to solve inherently serial problems," *arXiv preprint arXiv:2402.12875*, 2024.

[13] Y. Li, K. Sreenivasan, A. Giannou, D. Papailiopoulos, and S. Oymak, "Dissecting chain-of-thought: Compositionality through in-context filtering and learning," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[14] S. K. Choe, W. Neiswanger, P. Xie, and E. Xing, "BETTY: An automatic differentiation library for multilevel optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2023.

[15] S. Ghadimi and M. Wang, "Approximation methods for bilevel programming," *arXiv preprint arXiv:1802.02246*, 2018.

[16] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design," in *Proceedings of International Conference on Machine Learning (ICML)*, 2021, pp. 4882–4892.

[17] J. Li, B. Gu, and H. Huang, "A fully single loop algorithm for bilevel optimization without Hessian inverse," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, vol. 36, pp. 7426–7434.

[18] Z. Lu and S. Mei, "First-order penalty methods for bilevel optimization," *SIAM Journal on Optimization*, vol. 34, no. 2, pp. 1937–1969, 2024.

[19] S. Lu, "SLM: A smoothed first-order lagrangian method for structured constrained nonconvex optimization," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[20] R. Sato, M. Tanaka, and A. Takeda, "A gradient method for multilevel optimization," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 7522–7533.

[21] H. Shen and T. Chen, "A single-timescale analysis for stochastic approximation with multiple coupled sequences," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022, pp. 17415–17429.

[22] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[23] O. So and C. Fan, "Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[24] Y. Huang and Q. Lin, "Oracle complexity of single-loop switching subgradient methods for non-smooth weakly convex functional constrained optimization," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024, vol. 36.

[25] L. L. Gao, J. J. Ye, H. Yin, S. Zeng, and J. Zhang, "Moreau envelope based difference-of-weakly-convex reformulation and algorithm for bilevel programs," *arXiv preprint arXiv:2306.16761*, 2023.

[26] R. Liu, Z. Liu, W. Yao, S. Zeng, and J. Zhang, "Moreau envelope for nonconvex bi-level optimization: A single-loop and hessian-free solution strategy," in *Proceedings of International Conference on Machine Learning (ICML)*, 2024.

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[28] D. Kinga, J. B. Adam, et al., "A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*. San Diego, California;, 2015, vol. 5, p. 6.

[29] H. Li, M. Wang, S. Lu, X. Cui, and P.-Y. Chen, "Training nonlinear transformers for chain-of-thought inference: A theoretical generalization analysis," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

Authorized licensed use limited to: The Libraries at Rensselaer Polytechnic Institute. Downloaded on April 18,2025 at 22:00:56 UTC from IEEE Xplore. Restrictions apply.